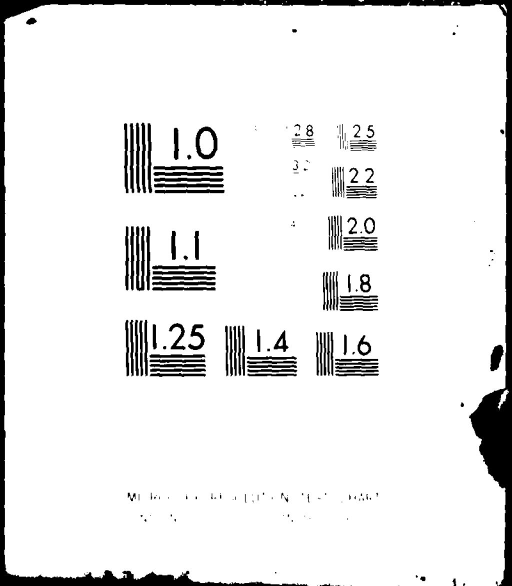
AD-A115 971 AIR FORCE ACADEMY CO F/G 9/2
SEVEN-HOLE PROBE DATA ACQUISITION SYSTEM.(U)
NOV 81 A GERNER, G SISSON
UNCLASSIFIED USAFA-TN-81-8 NL

1 OF 2

AD A
115971

1 OF 2

AD A
115971



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

USAFA-TN-81-8

Department of Aeronautics
Dean of the Faculty
United States Air Force Academy
Colorado 80840

SEVEN-HOLE PROBE DATA ACQUISITION SYSTEM

TECHNICAL NOTE
USAFA-TN-81-8

Gerner, A., et.al.
Sisson, G., et.al.

18 NOVEMBER 1981

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

Any views expressed in this paper are those of the author. They should not be interpreted as reflecting the views of the USAF Academy or the official opinion of any governmental agency. Notes are not reviewed for content or quality by the USAF Academy but are published primarily as a service to the faculty to facilitate internal research communication.

This Technical Note has been cleared for open publication and/or public release by the appropriate Office of Information in accordance with AFR 190-17 and DODD 5230.9. There is no objection to unlimited distribution of this Technical Note to the public at large or by DDC to the National Technical Information Service.

This Technical Note is approved for publication.

Clayton V Stewart

Clayton V. Stewart, Lt Colonel, USAF
Director of Research and Continuing Education

FORWARD

This technical note is the final report covering the period 1 Oct 80 to 30 Sep 81 in response to NASA A83011B sponsored by Nasa Ames Research Center and administered by Mr. Tom Gregory. Capt. G. Sisson was the principal investigator and was aided by staff and cadets at the Air Force Academy. This technical note consists of two separate papers. The first covers the compressible calibration of the seven-hole probe and is a reprint of an article in the Aeronautics Digest - Fall/Winter 1980, USAFA-TR-81-4. The second paper describes the associated software for the data acquisition system and contains program listings. Together, the two papers describe the entire data acquisition system developed to measure flow field properties quickly and economically in wakes using the seven-hole probes.

CALIBRATION OF SEVEN-HOLE PROBES
SUITABLE FOR HIGH ANGLES IN SUBSONIC COMPRESSIBLE FLOWS

A.A. Gerner and C.L. Maurer*

Abstract

This paper illustrates, by example, a method for calibrating seven-hole probes to measure local total and static pressures and relative flow angles of up to 70 degrees in subsonic compressible flows. To conserve air in our blowdown wind tunnel, we used the method of Latin Squares to statistically sample a large and otherwise unmanageable data set, thereby reducing to a minimum the number of data points required to construct a polynomial fit to the data. The three-variable third order polynomials found to represent the probe calibration permit all the desired output quantities to be found explicitly from pressures measured on the probe in an unknown flow field. This method determines the flow angle to within ±2 degrees with 95 percent certainty.

I. Introduction

Many present and future aircraft designs are beginning to employ such devices as leading-edge strakes, forward-swept wings, and canards. These devices have demonstrated the potential for enhancing aircraft maneuverability and control by producing strong vortices. Some modern aircraft, such as the Concorde, actually rely on vortices and complex flows to create lift However, in some instances, primary and secondary vortices can interact unfavorably, causing separation and loss of lift on portions of a wing. To be sure, these vortex-laden flow fields are quite intricate and difficult to analyze. Flow visualization techniques offer a way to gain insight into vortex interactions, but suffer from their inability to provide quantitative information. To overcome this limitation, small probes can be inserted directly into the flow stream to gather meaningful pressure information. Historically, non-nulling five-hole probes have been used to determine local total and static pressures at a particular point in a flow, as well as flow directions up to 40 degrees relative to their axis. Nevertheless, it is not inconceivable for the local flow angles of strong vortices to exceed 60 degrees. For this reason, the Air Force Academy, under a grant from NASA-Ames, has developed a unique seven-hole probe. In addition to local total and static pressure measurements, these probes have demonstrated the ability to determine flow angles up to 80 degrees relative to their axis. When combined with a computerized data acquisition system, they are capable of taking data at a rate of nearly two data points per second, much faster than nulling probes which require considerable time to balance probe tip pressures before each pressure measurement can be taken.

In addition, these probes are very small (about one-tenth of an inch in diameter), so they do not significantly disturb the flow they are measuring. But because of this small size, they suffer from inherent manufacturing defects. As a result, each probe must be calibrated before it can be a useful measuring device. Gallington describes such a calibration procedure (Ref. 1) which is both fast and effective, but is only valid

*Cadets, USAF Academy

for incompressible flows. This constraint effectively restricts the probes to surveying flows with free-stream Mach numbers of 0.3 and below (Ref. 2). However, the mere presence of a body in an airflow causes the flow to accelerate in certain regions, resulting in local velocities greater than the free-stream velocity. In addition, local flow velocities in vortex fields are likely to exceed those of the free stream too. As a result, even though the free-stream conditions are slow enough to justify an incompressible flow assumption, the local flow conditions might in fact be compressible. The purpose of this report then is to develop a power series calibration scheme which accurately determines the actual flow conditions from pressures measured on the probe in compressible flows.

Because of the similarities in compressible and incompressible theory, we begin our discussion by developing fully the incompressible calibration theory. Then, by analogy, we expand this theory to obtain the desired form of the calibration suitable for compressible flow. Next, we describe the apparatus and procedures we used to calibrate a probe in subsonic compressible flow, and finally, we discuss the results of that calibration.

II. Incompressible Flow Theory

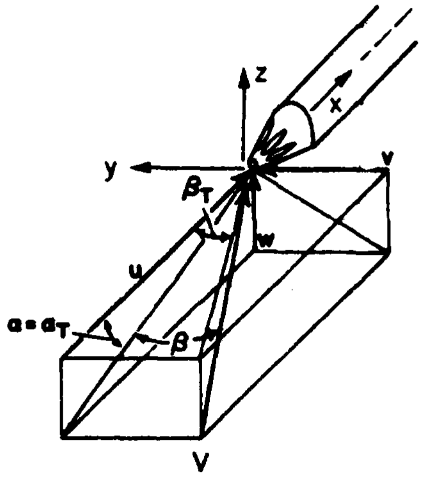
To simplify our discussion on the calibration of seven-hole probes in compressible flow, we will begin with a background in the theory of incompressible flow calibrations. Much of the information presented in the following sections is drawn from the work of Gallington (Ref. 1). Readers wishing further information on the theory of incompressible probe calibrations are highly encouraged to consult this reference.

Axis System for Low Flow Angles

To begin our discussion of calibration theory, we restrict our treatment to low flow angles; typically, those values for which the angle between the velocity vector of the flow stream and the probe's axis are less than 30 degrees. The more familiar reference system for low flow angles measures velocity vectors in terms of the angle of attack, α, and the angle of sideslip, β. However, in choosing our reference system, we adopted the tangential reference system illustrated in Figure 1. In this system, α_T is taken to be the projection on the vertical plane of the angle between the velocity vector and the probe's axis. And to preserve symmetry, β_T is defined as the projection on the horizontal plane of the angle between the probe's axis and the relative wind. For this reason, the tangential reference system differs slightly from the α-β system, but will be used to evaluate the low angle flow properties.

Pressure Coefficients for Low Flow Angles

For low flow angles it is desirable to define dimensionless pressure coefficients which utilize all seven measured probe pressures and are sensitive to changes in flow angularity with respect to the probe's x-axis. From Figure 2, one such pressure coeffi-



CONVENTIONAL	TANGENT
$u = V \cos\alpha \cos\beta$	$\alpha_T = \arctan \frac{w}{u}$
$v = V \sin\beta$	$\beta_T = \arctan \frac{v}{u}$
$w = V \sin\alpha \cos\beta$	

Figure 1. Low Angle Reference System

cient sensitive to changes in angle of attack in the x-z planes is defined as:

$$C_{\alpha_1} = \frac{P_4 - P_1}{P_7 - \overline{P}_{1-6}} \tag{1}$$

where the numerator measures changes in flow angularity based on the differences in opposite port pressures, and the denominator nondimensionalizes the term with the apparent dynamic pressure. This pseudo-dynamic pressure is obtained from the difference between the central port pressure, P_7, which approximates the total pressure at low angles, and the average of the six surrounding pressures, $\overline{P}_{1-6}$, which collectively approximates the static pressure. From the definition of this pressure coefficient, it is easy to see that two other possibilities also exist: one which measures the pressure differential

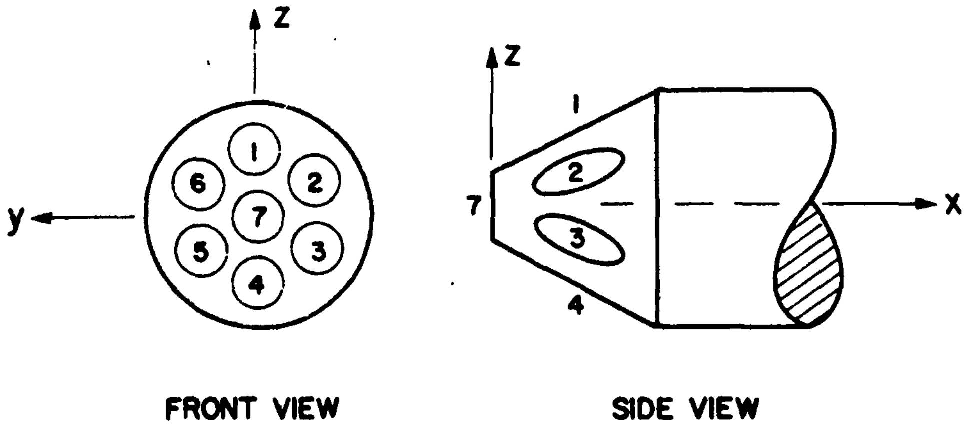


Figure 2. Port Numbering Convention and Principal Axes

between ports three and six, and the other, which measures the pressure differential between ports two and five. The complete set of these pressure coefficients include:

$$C_{\alpha_1} = \frac{P_4 - P_1}{P_7 - \bar{P}_{1-6}} \quad , \quad C_{\alpha_2} = \frac{P_3 - P_6}{P_7 - \bar{P}_{1-6}} \quad , \quad C_{\alpha_3} = \frac{P_2 - P_5}{P_7 - \bar{P}_{1-6}} \qquad (2)$$

But before these coefficients can be of any use to us, they must be resolved into the α_T - β_T reference system. This is done by weighing the contribution of each coefficient in Eqns. (2) along the respective axis, which results in the following two equations:

$$C_\alpha = \frac{1}{3}\,(2C_{\alpha_1} + C_{\alpha_2} - C_{\alpha_3}) \quad , \quad C_\beta = \frac{1}{\sqrt{3}}\,(C_{\alpha_2} + C_{\alpha_3}) \qquad (3)$$

The first equation defining C_α contains all three coefficients of Eqns. (2). In particular, C_{α_1} has the greatest significance, which only makes sense since it lies directly along the axis of interest. The equation defining C_β only takes into account the last two coefficients of Eqns. (2), assigning to each an equal significance. The fact that C_{α_1} is not included in this equation again makes sense, since it is directly aligned with the α_T-direction, ideally making it insensitive to changes in the perpendicular β_T-direction. In summary, the procedure of obtaining C_α and C_β requires two tasks. First, determine C_{α_1}, C_{α_2}, and C_{α_3} from the seven measured pressures using Eqns. (2) and then substitute these intermediate quantities into Eqns. (3) for the desired coefficients.

Having defined the two angular pressure coefficients, it is now appropriate to discuss the remaining low angle pressure coefficients, C_o and C_q, defined as:

$$C_o = \frac{P_7 - P_{oL}}{P_7 - \bar{P}_{1-\epsilon}} \quad , \quad C_q = \frac{P_7 - \bar{P}_{1-6}}{P_{oL} - P_{\infty L}} \qquad (4)$$

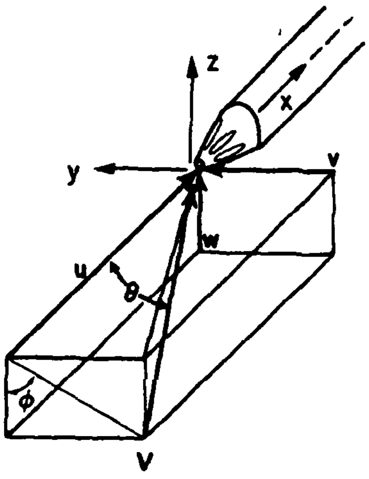
C_o is the apparent total pressure coefficient and functions as a correction factor to convert actual pressures measured by the probe to accurate values of local total pressure. From the numerator, it is seen that the coefficient measures the difference between the pseudo-total pressure measured by the probe, P_7, and the actual total pressure. Just as with Eqn. (1), the coefficient is nondimensionalized by the denominator, which is a measure of the apparent dynamic pressure as previously described.

C_q functions much like C_o, but instead of correcting probe pressures to total pressure, C_q relates these pressures to the actual dynamic preosure. In this coefficient, the numerator represents the probe's approximation of dynamic pressure while the denominator consists of the actual dynamic pressure.

Axis System for High Flow Angles

Up to this point, our discussion has been limited to a description of the pressure coefficients used for flow angles below 30 degrees. Yet, the real advantage of using a seven-hole probe lies in its ability to determine flow angles as high as 80 degrees to the probe's x-axis.

A reference system better suited to high angle measurement than the tangential system is the polar reference system, which measures flow angularities in terms of θ and ϕ and is shown in Figure 3. In this system, θ, the pitch angle, is the angle the velocity vector makes with respect to the probe's x-axis; and ϕ, the roll angle, describes the azmuthal orientation of the velocity vector in the y-z plane, measured counterclockwise from the negative z-axis as viewed from the front. Although a singularity exists directly along the x-axis, this does not represent any problems in high angle measurement and is avoided entirely by switchihg to the tangential reference system at low angles.



POLAR	TANGENT
$u = V \cos\theta$ $v = V \sin\theta \sin\phi$ $w = V \sin\theta \cos\phi$	$\alpha_T = \arctan \frac{w}{u}$ $\beta_T = \arctan \frac{v}{u}$

Figure 3. High Angle Reference System

Pressure Coefficients for High Flow Angles

At low angles of attack, all seven of the measured probe pressures are used to form the pressure coefficients. However, at high angles of attack, as illustrated in Figure 4, the flow tends to detach over the downstream portions of the probe. Pressure ports lying in this separated region are insensitive to changes in flow angularity; consequently, it is not feasible to use their pressures in a meaningful coefficient. As a result, the pressure coefficients for high angle measurements must be defined so that they include only the pressures from ports in attached flow.

Typically, the separation points of a cylinder in turbulent flow are over 100 degrees from the frontal stagnation point (Ref. 3). And for a conical body, such as the probe tip,

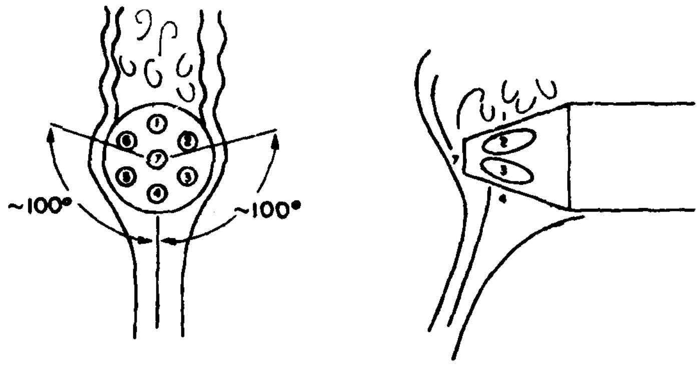


Figure 4. Flow Over Probe at High Angle of Attack

the flow is likely to remain attached longer. In addition, the u-velocity component is also likely to extend the separation points. Thus, for the condition depicted in Figure 4, pressure ports three, four, five, and seven lie in reliably attached flow; port one is in separated flow; and the disposition of ports two and six is uncertain. Using only those ports in attached flow, a coefficient sensitive to the angle of pitch is defined as:

$$C_{\theta_4} = \frac{P_4 - P_7}{P_4 - \dfrac{P_3 + P_5}{2}} \qquad (5)$$

Following the same rationale as in the low angle case, the numerator measures changes in θ based on the differences in opposing port pressures. In this example, P_7, the smaller of the two pressures, is subtracted from P_4. Here again, the coefficient is nondimensionalized by dividing through with the apparent dynamic pressure. This pseudo-dynamic pressure is determined from the difference between the peripheral port pressure, P_4, which at high angles approximates the total pressure, and the average of P_3 and P_5, which when taken together are relatively independent to changes in roll and approximate the static pressure.

Using a similar argument, a coefficient which changes in proportion to roll angle is appropriately defined:

$$C_{\phi_4} = \frac{P_3 - P_5}{P_4 - \dfrac{P_3 + P_5}{2}} \qquad (6)$$

The numerator of Eqn. (6) is sensitive to changes in ϕ, in that as the velocity vector rolls in either direction, the windward pressure rises and the leeward pressure falls. In this way, the difference between the two pressures varies significantly for variations in roll, yet the average of their sums remains relatively constant. Once more, the coefficient is nondimensionalized with the same denominator as in the previous case.

Obviously, the above two coefficients are only valid for a narrow range in roll angle about port four. That is, as we rotate the velocity vector to either side of port four, the region of separated flow approaches either port five or three. Therefore, to insure that all pressures are taken from ports in attached flow, we restrict Eqns. (5) and (6) to roughly a 60-degree pie-shaped sector centered on port four. In this way, six pie-shaped sectors are summarily defined for high angle measurement, such that each has its own set of coefficients based on the pressures in attached flow. The remaining angular pressure coefficients are defined with the same method used to develop Eqns. (5) and (6), resulting in the following set of equations:

$$
\begin{aligned}
C_{\theta_1} &= \frac{P_1 - P_7}{P_1 - \dfrac{P_2 + P_6}{2}} , & C_{\phi_1} &= \frac{P_6 - P_2}{P_1 - \dfrac{P_6 + P_2}{2}} \\
C_{\theta_2} &= \frac{P_2 - P_7}{P_2 - \dfrac{P_1 + P_3}{2}} , & C_{\phi_2} &= \frac{P_1 - P_3}{P_2 - \dfrac{P_1 + P_3}{2}} \\
C_{\theta_3} &= \frac{P_3 - P_7}{P_3 - \dfrac{P_2 + P_4}{2}} , & C_{\phi_3} &= \frac{P_2 - P_4}{P_3 - \dfrac{P_2 + P_4}{2}} \\
C_{\theta_4} &= \frac{P_4 - P_7}{P_4 - \dfrac{P_3 + P_5}{2}} , & C_{\phi_4} &= \frac{P_3 - P_5}{P_4 - \dfrac{P_3 + P_5}{2}} \\
C_{\theta_5} &= \frac{P_5 - P_7}{P_5 - \dfrac{P_4 + P_6}{2}} , & C_{\phi_5} &= \frac{P_4 - P_6}{P_5 - \dfrac{P_4 + P_6}{2}} \\
C_{\theta_6} &= \frac{P_6 - P_7}{P_6 - \dfrac{P_5 + P_1}{2}} , & C_{\phi_6} &= \frac{P_5 - P_1}{P_6 - \dfrac{P_5 + P_1}{2}}
\end{aligned}
\tag{7}
$$

Similarly, the C_O and C_q coefficients are developed with the same rationale used to derive their low angle counterparts; the only difference resides in the choice of the pressures which are roughly equivalent to total and static pressures. These pressures, of course, vary in relation to the sector a particular coefficient describes. The complete set of these coefficients include:

$$\begin{array}{llll}
C_{o1} = \dfrac{P_1 - P_{oL}}{P_1 - \dfrac{P_2 + P_6}{2}} & , & C_{q1} & \dfrac{P_1 - \dfrac{P_2 + P_6}{2}}{P_{oL} - P_{\infty L}} \\[2em]
C_{o2} = \dfrac{P_2 - P_{oL}}{P_2 - \dfrac{P_3 + P_1}{2}} & , & C_{q2} & \dfrac{P_2 - \dfrac{P_3 + P_1}{2}}{P_{oL} - P_{\infty L}} \\[2em]
C_{o3} = \dfrac{P_3 - P_{oL}}{P_3 - \dfrac{P_4 + P_2}{2}} & , & C_{q3} & \dfrac{P_3 - \dfrac{P_4 + P_2}{2}}{P_{oL} - P_{\infty L}} \\[2em]
C_{o4} = \dfrac{P_4 - P_{oL}}{P_4 - \dfrac{P_5 + P_3}{2}} & , & C_{q4} & \dfrac{P_4 - \dfrac{P_5 + P_3}{2}}{P_{oL} - P_{\infty L}} \\[2em]
C_{o5} = \dfrac{P_5 - P_{oL}}{P_5 - \dfrac{P_6 + P_4}{2}} & , & C_{q5} & \dfrac{P_5 - \dfrac{P_6 + P_4}{2}}{P_{oL} - P_{\infty L}} \\[2em]
C_{o6} = \dfrac{P_6 - P_{oL}}{P_6 - \dfrac{P_1 + P_5}{2}} & , & C_{q6} & \dfrac{P_6 - \dfrac{P_1 + P_5}{2}}{P_{oL} - P_{\infty L}}
\end{array} \qquad (8)$$

Division of Angular Space

Having defined a host of coefficients for low and high angles and for various sectors around the probe, a question arises as to when a particular set of coefficients should be used. To arbitrarily assign angular cut-offs, based on probe symmetries, would be naive since actual data might suggest better division lines. With this in mind, a better scheme for locating the sector division lines is based on the isobars depicted in Figure 5. This method defines seven sectors, the central low angle sector and the six high angle periphery sectors, and allocates data points to a given sector based on the highest port pressure measured on the probe.

Polynomial Power Series Expansion

Once the data points are allocated to the proper sector with its corresponding pressure coefficients, a fourth order polynomial expansion is used to solve for the desired quantities. In two variables (i.e., using the two angular pressure coefficients) this

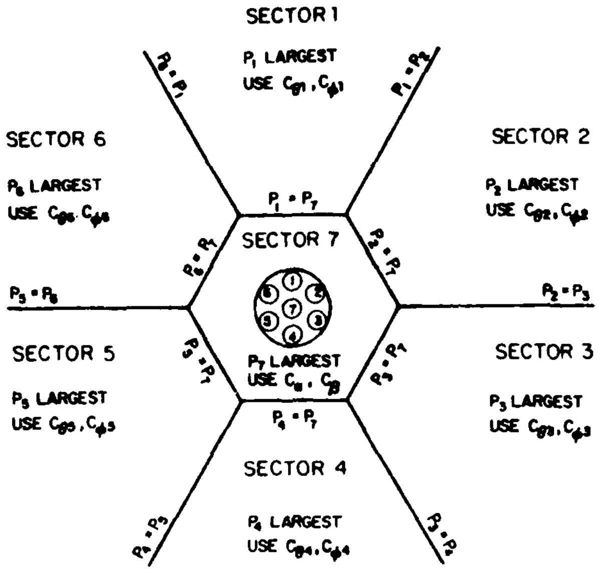


Figure 5. Division of Angular Space

expansion takes on the following form:

$$
\begin{array}{ll}
 & \underline{\text{Order of Terms}} \\
A_i \approx K_1^A + & \text{0th} \\
\quad K_2^A C_{\alpha_i} + K_3^A C_{\beta_i} + & \text{1st} \\
\quad K_4^A C_{\alpha_i}^2 + K_5^A C_{\alpha_i} C_{\beta_i} + K_6^A C_{\beta_i}^2 + & \text{2nd} \\
\quad K_7^A C_{\alpha_i}^3 + K_8^A C_{\alpha_i}^2 C_{\beta_i} + K_9^A C_{\alpha_i} C_{\beta_i}^2 + K_{10}^A C_{\beta_i}^3 + & \text{3rd} \\
\quad K_{11}^A C_{\alpha_i}^4 + K_{12}^A C_{\alpha_i}^3 C_{\beta_i} + K_{13}^A C_{\alpha_i}^2 C_{\beta_i}^2 + K_{14}^A C_{\alpha_i} C_{\beta_i}^3 + K_{15}^A C_{\beta_i}^4 & \text{4th}
\end{array} \tag{9}
$$

where A is either α_T, β_T, C_o or C_q for low angles and θ, ϕ, C_o or C_q for high angles, with the subscript denoting the ith such quantity. The K's are the calibration coefficients, with the superscripts denoting the quantity to which a particular set of K's belong, and the subscripts identifying the coefficient of a particular term in the power series expansion. Note that in the high angle case, the C_α's and C_β's are replaced by C_θ's and C_ϕ's respectively. In matrix notation for "n" data points of a particular sector, a set of Eqn. (9)'s are represented as:

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \cdot \\ \cdot \\ \cdot \\ A_n \end{bmatrix} = \begin{bmatrix} 1 & C_{\alpha_1} & C_{\beta_1} & C^2_{\alpha_1} & C_{\alpha_1}C_{\beta_1} & C^2_{\beta_1} & \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot & C^4_{\beta_1} \\ 1 & C_{\alpha_2} & C_{\beta_2} & C^2_{\alpha_2} & C_{\alpha_2}C_{\beta_2} & C^2_{\beta_2} & \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot & C^4_{\beta_2} \\ 1 & C_{\alpha_3} & C_{\beta_3} & C^2_{\alpha_3} & C_{\alpha_3}C_{\beta_3} & C^2_{\beta_3} & \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot & C^4_{\beta_3} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot \\ 1 & C_{\alpha_n} & C_{\beta_n} & C^2_{\alpha_n} & C_{\alpha_n}C_{\beta_n} & C^2_{\beta_n} & \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot & C^4_{\beta_n} \end{bmatrix} \begin{bmatrix} K^A_1 \\ K^A_2 \\ K^A_3 \\ \cdot \\ \cdot \\ \cdot \\ K^A_{15} \end{bmatrix} \qquad (10)$$

But to simplify further discussion, Eqn. (10) is rewritten as:

$$[A] = [C][K] \qquad (11)$$

In this form, the n x 1 A-matrix contains n values of the flow parameters of interest, the n x 15 C-matrix contains the expanded angular pressure coefficients for each of the n flow parameters, and the 15 x 1 K-matrix contains the calibration coefficients for the particular flow parameter of interest.

Determining the Calibration Coefficients

During the calibration process, the quantities within the A-matrix are set up by the experimenter in terms of the known tunnel conditions, and the terms within the C-matrix are determined from the measured probe pressures. A calibration procedure, therefore, involves the calculation of the unknown K-matrix. This calculation is performed by rearranging Eqn. (11) to solve for the unknown calibration coefficients. With matrix algebra, this is performed according to the procedure outlined by Netter and Wasserman (Ref. 4):

First multiply each side of Eqn. (11) by the transpose of the C-matrix:

$$[C]^T\ [A] = [C]^T\ [C][K] = [C^TC][K] \qquad (12)$$

Next multiply each side by the inverse of the recently created C^TC-matrix:

$$[C^TC]^{-1}[C]^T[A] = [C^TC]^{-1}[C^TC][K] \qquad (13)$$

Realizing that the product of a matrix and its inverse results in the identity matrix, Eqn. (13) simplifies to yield a solution for the unknown K-matrix in terms of the known C- and A-matrices:

$$[K] = [C^TC]^{-1}[C]^T[A] \tag{14}$$

This technique determines the calibration coefficients by a least squares curve fit to the experimental data.

Determining the Desired Flow Propcrties

Once the calibration coefficients are determined, the calibration process is complete. The probe is ready to be insertec in an unknown flow field and the desired flow properties determined. Once in the flow field, the probe's measured pressure readings allow us to determine the angular pressure coefficients; these coefficients are then manipulated to fill the C-matrix of Eqn. (10). Since the K-matrix is already known, the desired flow properties in the A-matrix are then determined explicitly. For a particular condition, the solutions for the desired flow properties take on the following functional forms:

Inner Sector (low flow angles)

$$\begin{aligned}
\alpha_T &= f(C_\alpha, C_\beta) = K_1^{\alpha_T} + K_2^{\alpha_T} C_\alpha + K_3^{\alpha_T} C_\beta + \dots\dots + K_{15}^{\alpha_T} C_\beta^4 \\
\beta_T &= f(C_\alpha, C_\beta) = K_1^{\beta_T} + K_2^{\beta_T} C_\alpha + K_3^{\beta_T} C_\beta + \dots\dots + K_{15}^{\beta_T} C_\beta^4 \\
C_o &= f(C_\alpha, C_\beta) = K_1^{C_o} + K_2^{C_o} C_\alpha + K_3^{C_o} C_\beta + \dots\dots + K_{15}^{C_o} C_\beta^4 \\
C_q &= f(C_\alpha, C_\beta) = K_1^{C_q} + K_2^{C_q} C_\alpha + K_3^{C_q} C_\beta + \dots\dots + K_{15}^{C_q} C_\beta^4
\end{aligned} \tag{15}$$

Outer Sectors (high flow angles)

$$\begin{aligned}
\theta &= f(C_{\theta_n}, C_{\phi_n}) = K_1^\theta + K_2^\theta C_{\theta_n} + K_3^\theta C_{\phi_n} + \dots\dots + K_{15}^\theta C_{\phi_n}^4 \\
\phi &= f(C_{\theta_n}, C_{\phi_n}) = K_1^\phi + K_2^\phi C_{\theta_n} + K_3^\phi C_{\phi_n} + \dots\dots + K_{15}^\phi C_{\phi_n}^4 \\
C_{o_n} &= f(C_{\theta_n}, C_{\phi_n}) = K_1^{C_{o_n}} + K_2^{C_{o_n}} C_{\theta_n} + K_3^{C_{o_n}} C_{\phi_n} + \dots + K_{15}^{C_{o_n}} C_{\phi_n}^4 \\
C_{q_n} &= f(C_{\theta_n}, C_{\phi_n}) = K_1^{C_{q_n}} + K_2^{C_{q_n}} C_{\theta_n} + K_3^{C_{q_n}} C_{\phi_n} + \dots + K_{15}^{C_{q_n}} C_{\phi_n}^4
\end{aligned} \tag{16}$$

Once the local total and dynamic pressure coefficients are specified, it is possible to determine the local total and dynamic pressures. This is accomplished by rearranging Eqns. (4) in the case of the inner sector, and Eqns. (8) in the case of the outer sectors, thereby solving for the desired pressures. This calculation involves the polynomial result for C_o or C_q and the seven measured probe pressures. As an example, the inner sector equations for local total and dynamic pressures are derived as follows:

Recalling that

$$C_o = \frac{P_7 - P_{oL}}{P_7 - \overline{P}_{1-6}} \quad \text{and} \quad C_q = \frac{P_7 - \overline{P}_{1-6}}{P_{oL} - P_{\infty L}} \tag{4}$$

the local total and dynamic pressures are then solved by manipulating Eqns. (4); thus:

$$P_{oL} = P_7 - C_o\,(P_7 - \overline{P}_{1-6}) \qquad P_{oL} - P_{\infty L} = \frac{P_7 - \overline{P}_{1-6}}{C_q} \tag{17}$$

The same procedure is extended to determine the local total and dynamic pressures in the outer sectors.

III. Extension to Compressible Flow

Up to this point we have discussed the form of a calibration procedure which is only valid for incompressible flow. This is so because the calibration coefficients depend directly upon the angular pressure coefficients, which as pressure coefficients are themselves dependent upon Mach number (*Ref. 5*). As such, seven-hole probes with the present method of calibration are limited to surveying flows within the incompressible regime. This restriction is lifted easily enough with the inclusion of an additional pressure coefficient representative of compressibility. Consequently, the number of variables in the polynomial curve fit is increased from two (for incompressible flow) to three (for compressible flow).

Determining a Coefficient of Compressibility

Just like the angular pressure coefficients (C_α and C_β or C_θ and C_ϕ), the compressibility coefficient (hereafter denoted C_M) must be nondimensional and determined strictly from pressures measured on the probe. In addition, C_M must have the feature that it approaches zero at very low Mach numbers. That is, at very low Mach numbers all terms bearing a compressibility coefficient should be insignificant in the power series expansion. This, in effect, brings us back to an incompressible power series expansion in two variables, where the significant terms only contain the angular pressure coefficients. A further constraint requires the compressibility coefficient to approach a finite value in the hypersonic limit. In other words, for very high Mach numbers, large changes in Mach number should have a neglible effect on the compressibility coefficient. This reflects a limitation shared by all pressure probe methods in that Mach number becomes indeterminate at hypersonic speeds (Ref. 1). These requirements are satisfied by modeling the compressibility coefficient after the dynamic to total pressure ratio depicted in Figure 6. This figure is idealized for isentropic flow; as such, it is not useful for flow speeds much beyond Mach one. However, since our calibration is limited to subsonic speeds, the isentropic idealization is valid through the sonic limit.

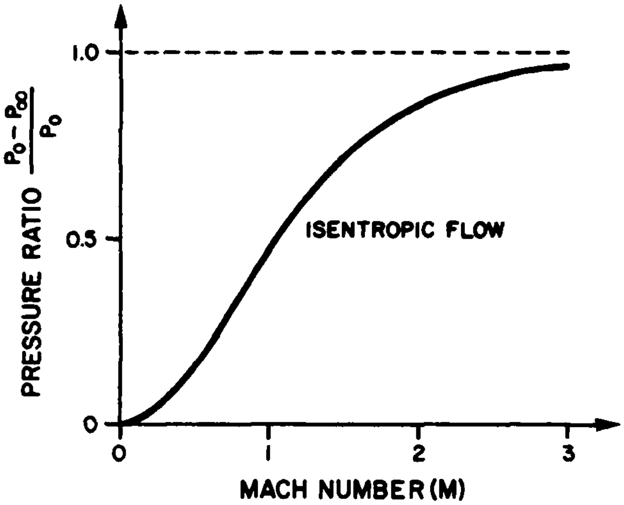


Figure 6. Compressibility Effects as a Function of Mach Number

To develop a compressibility coefficient from probe measured pressures, we need to represent the total and dynamic pressures from probe measured pressures. For the low angle case, a pseudo-total pressure is approximated by P_7 with a pseudo-dynamic pressure approximated by $P_7 - \overline{P}_{1-6}$. Thus, for the inner sector, a compressibility coefficient in terms of the dynamic to total pressure ratio is modeled by:

$$C_{M_7} = \frac{P_7 - \overline{P}_{1-6}}{P_7} \tag{18}$$

The compressibility coefficients for each of the outer sectors are modeled in a similar manner; these outer sector coefficients include:

$$C_{M_1} = \frac{P_1 - \frac{P_6 + P_2}{2}}{P_1}$$

$$C_{M_2} = \frac{P_2 - \frac{P_1 + P_3}{2}}{P_2} \tag{19}$$

$$C_{M_3} = \frac{P_3 - \frac{P_2 + P_4}{2}}{P_3}$$

$$C_{M_4} = \frac{P_4 - \dfrac{P_3 + P_5}{2}}{P_4}$$

$$C_{M_5} = \frac{P_5 - \dfrac{P_4 + P_6}{2}}{P_5} \qquad (19)$$

$$C_{M_6} = \frac{P_6 - \dfrac{P_5 + P_1}{2}}{P_6}$$

Selection of Data Points

Typical incompressible probe calibrations take approximately 80 data points in two variables (C_α and C_β) for each of the seven sectors (Ref. 1). This results in a total of about 560 data points for a complete calibration. To extend this present scheme into yet another dimension (Mach number) would create a data set of intractable proportions. Consequently, it is necessary to represent the data set with a sample of more manageable proportions. In addition, this sample must be chosen such that the density of chosen data points throughout the data set is homogenous. In other words, the sample must be an accurate representation of the data set, otherwise the calibration routine will not offer consistent accuracy throughout the range of data.

A method of ensuring a homogenous, yet random, sample of a three-dimensional parameter space is suggested by Cochran and Cox (Ref. 6). The technique is known as the method of Latin Squares, an example of which is shown in Figure 7. The plan depicted in this figure is a 3 x 3 square which actually represents a three-dimensional parameter space; one variable along the vertical axis, a second variable along the horizontal axis, and the third variable denoted by the letters along the axis going into the page. This square is better visualized in Figure 8, where it is shown in three dimensions instead

A	B	C
B	C	A
C	A	B

Figure 7. 3 x 3 Latin Square Plan

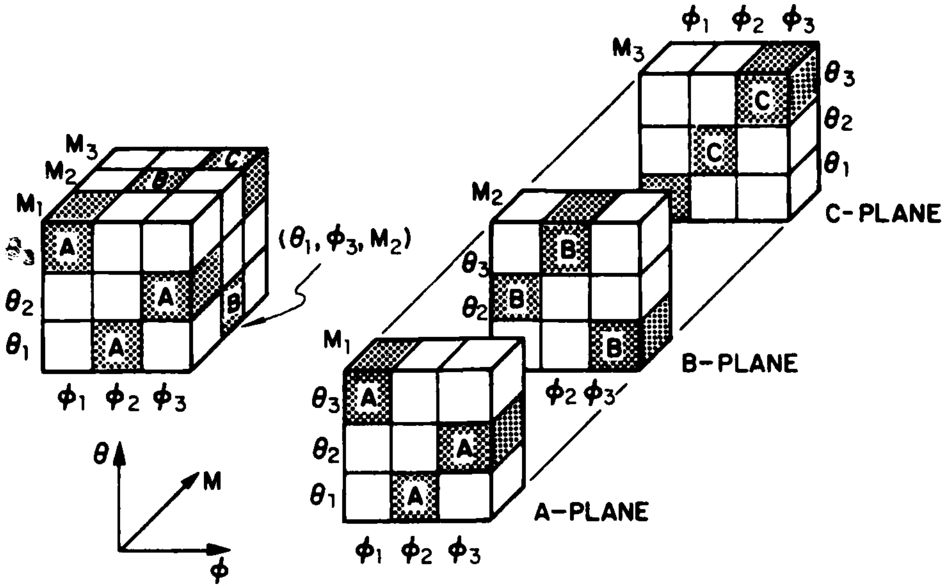


Figure 8. 3 x 3 Latin Square Shown in Three Dimensions

of the two-dimensional rendition of Figure 7. The 3 x 3 x 3 cube represents the entire parameter space, and each sub-cube represents a discrete data point within the entire data set. In Figure 7, the A's denote the data points to be selected in the A-plane (i.e., the plane in which the third variable, Mach number, is held constant), the B's denote the points selected in the B-plane, and so on. In this way, the entire data set is sampled by the points which appear as the shaded cubes in Figure 8. The unique feature of this sampling technique is that no matter from which axis direction the cube is viewed, the entire frontal space will appear covered by data points, one point deep; that is, no two points will appear to overlap each other. Furthermore, in any given plane where one variable is held constant, every value of the remaining two variables is sampled exactly once. Consequently, the method of Latin Squares guarantees a homogenous density of data points comprising the sample of the representative data set. Of course, the technique of Latin Squares is not limited to a 3 x 3 plan, but can be expanded to a 12 x 12 plan if desired; however, the more common plans range between the 5 x 5 and 8 x 8 squares (Ref. 6). This is true because as plan size increases, the ratio of points sampled to total points in the data set decreases. To illustrate, the 3 x 3 square of Figure 7 samples a 27-point data set with nine points, whereas a 10 x 10 square uses 100 points to sample a 1000-point set. Accordingly, the 3 x 3 Latin Square samples 33 percent of the entire data set, while the 10 x 10 Latin Square samples only 10 percent of the complete set. Nevertheless, a 3 x 3 square describes a given parameter space with only 27 data points, while a 10 x 10 square divides the same parameter space into

1000 discrete data points. Thus, a larger Latin Square represents a given parameter space with greater resolution, but samples the resulting data set with a smaller percentage of points. The experimenter, therefore, is faced with a compromise in choosing the Latin Square best suited to his needs.

For reasons to be discussed later, we selected a 6 x 6 Latin Square (shown in Figure 9) for each sector. This gives us a data set with six different values for each of the three variables θ, ϕ, and M, for a total of 216 data points per sector. Using the sampling technique of Latin Squares, each 216-point data set is represented by a sample of 36 points. As such, a total of 252 data points will be experimentally tested in all seven sectors for a complete calibration of the probe.

A	B	C	D	E	F
B	F	D	C	A	E
C	D	E	F	B	A
D	A	F	E	C	B
E	C	A	B	F	D
F	E	B	A	D	C

Figure 9. 6 x 6 Latin Square Plan

Polynomial Power Series Expansion in Three Variables

In the incompressible calibration, a fourth order polynomial expansion in two variables was used for a total of 15 terms with 15 corresponding calibration coefficients. By adding a third variable to the calibration, the number of terms in the fourth order expansion jumps from 15 to 35 with 35 corresponding calibration coefficients. In order to obtain a valid estimation of standard deviation, a surplus of about 20 data points over the number of calibration coefficients is necessary (Ref. 1). This sets the required number of data points for a given sector at 55; however, our sample size has already been constrained to 36 points due to the selection of a 6 x 6 Latin Square. It is possible to take more than one 36-point Latin Square sampling within the 216-point data set, but in the interest of keeping the total number of points required for a complete calibration down to a manageable number, so that time spent in the wind tunnel can be minimized, we elect to limit each of the seven samples to 36 data points. As a result, a fourth order curve fit is no longer feasible; consequently, the polynomial expansion is reduced to the next lowest order.

A third order polynomial expansion in three variables requires 20 calibration coefficients. This leaves us with a 16 data point surplus, which is sufficient to calculate a valid standard deviation. Using the same format at Eqn. (9), a third order expansior. in three variables takes on the following form:

$$
\begin{array}{ll}
 & \underline{\text{Order of Terms}} \\
A_i = K_1^A + & \text{0th} \\
\quad K_2^A C_{\alpha_i} + K_3^A C_{\beta_i} + K_4^A C_{M_i} + & \text{1st} \\
\quad K_5^A C_{\alpha_i}^2 + K_6^A C_{\beta_i}^2 + K_7^A C_{M_i}^2 + K_8^A C_{\alpha_i} C_{\beta_i} + K_9^A C_{\alpha_i} C_{M_i} + K_{10}^A C_{\beta_i} C_{M_i} + & \text{2nd} \\
\left.\begin{array}{l} \quad K_{11}^A C_{\alpha_i}^3 + K_{12}^A C_{\beta_i}^3 + K_{13}^A C_{M_i}^3 + K_{14}^A C_{\alpha_i}^2 C_{\beta_i} + K_{15}^A C_{\alpha_i}^2 C_{M_i} + \\ \quad K_{16}^A C_{\alpha_i} C_{\beta_i}^2 + K_{17}^A C_{\beta_i}^2 C_{M_i} + K_{18}^A C_{\alpha_i} C_{M_i}^2 + K_{19}^A C_{\beta_i} C_{M_i}^2 + K_{20}^A C_{\alpha_i} C_{\beta_i} C_{M_i} \end{array}\right\} & \text{3rd}
\end{array}
\qquad (20)
$$

where A is either α_T, β_T, C_o, or C_q for the inner sector and θ, ϕ, C_o, or C_q tor the outer sectors, with the subscript denoting the ith such quantity. The K's are the calibration coefficients, where the superscript denotes the quantity to which a particular set of K's belong, and the subscripts identify the coefficient of a particular term in the power series expansion. In the case of an outer sector, the $C\alpha$'s and $C\beta$'s are replaced by $C\epsilon$'s and $C\phi$'s, respectively.

Determination of Mach Number

As stated previously, this experiment limits its scope to surveying Mach numbers slightly below sonic flow on down to incompressible flow. Consequently, there are no shocks ahead of the probe and the isentropic flow relation applies:

$$\frac{P_\infty}{P_o} = \left[1 + \frac{\gamma-1}{2} M^2\right]^{\frac{-\gamma}{\gamma-1}} \qquad (21)$$

Rearranging Eqn. (21) and deriving Mach number in terms of the dynamic to total pressure yields:

$$M = \sqrt{\frac{2}{\gamma-1}\left(\left[1 - \frac{P_o - P_\infty}{P_o}\right]^{\frac{1-\gamma}{\gamma}} - 1\right)} \qquad (22)$$

And since we know C_o, C_q, and the seven probe pressures, we can explicitly determine the local Mach number from the dynamic to total pressure ratio. For the inner sector, this ratio is determined from Eqns. (4) and the seven probe pressures:

$$\frac{P_{oL} - P_{\infty L}}{P_{oL}} = \left[C_q \left(\frac{P_7}{P_7 - \overline{P}_{1-6}} - C_o\right)\right]^{-1} \tag{23}$$

Analagous equations for the outer sectors are developed in the same way, except that Eqns. (8) and the appropriate probe pressures are used instead. During calibration, the actual values of Eqn. (23) and its outer sector counterparts are known from the measured tunnel conditions. Solving Eqn. (21) in terms of the dynamic to total pressure ratio as a function of Mach number gives us:

$$\frac{P_{oL} - P_{\infty L}}{P_{oL}} = 1 - \left[1 + \frac{\gamma - 1}{2} M_L^2\right]^{\frac{\gamma}{1-\gamma}} \tag{24}$$

Thus, if the total and static pressures cannot be measured directly, we can still determine the dynamic to total pressure ratio if only the Mach number is known. By noting the difference between Eqns. (23) and (24), we can estimate how accurately our polynomial curve fit determines the Mach number for the inner sector; similar arguments are also extended to the outer sectors.

<u>Estimating Accuracy of the Curve Fit</u>

Providing there are approximately twenty more data points than the number of calibration coefficients, it is statistically feasible to calculate a global estimate of the accuracy of the curve fit for each of the flow parameters. This is done by computing the standard deviation of the difference between the experimental data and the polynomial prediction of that data. For flow angles and Mach number, the following relation applies (Ref. 7):

$$\sigma(A) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (A_{EXP_i} - A_{POLY_i})^2} \tag{25}$$

where σ is the standard deviation; n, the total number of data points; and A, the desired flow parameter.

Even though the total and dynamic pressures are determined from C_o and C_q, the accuracies of these pressures are not representative of the accuracies obtained for C_o and C_q. To estimate the uncertainty in determining the total and dynamic pressures (for the inner sector, for example) requires the defining equations, Eqns. (17), and applying to them the method of Kline and McClintock (Ref. 8). Then, by taking the standard deviation of these uncertainties and nondimensionalizing them with the dynamic pressure, we arrive at the following estimates for the accuracies of these pressures:

$$\frac{\sigma(P_{oL})}{P_{oL} - P_{\infty L}} \approx \bar{C}_{q_n} \sigma(C_o) \qquad (26)$$

$$\frac{\sigma(P_{oL} - P_{\infty L})}{P_{oL} - P_{\infty L}} \approx \frac{\sigma(C_q)}{\bar{C}_{q_n}} \qquad (27)$$

where $\bar{C}_{q_n}$ is the average value of this coefficient for a given sector denoted by "n".

Providing the errors between the actual values and the polynomial predictions of those values are normally distributed, there is a 68.3 percent probability that the polynomial prediction will fall within one standard deviation of the actual value. And at 1.96σ, this certainty is increased to 95 percent.

IV. Apparatus and Calibration Procedure

Probe Geometry

The seven-hole probe was constructed at the Air Force Academy by packing seven properly-sized stainless steel tubes into a larger stainless steel tube as shown in Figure 10. Although the inner seven tubes have an outside diameter of only 0.028 inches with a .005-inch wall thickness, accurate alignment is reasonably insured since the tubes can only pack in one unique way. Once assembled in this arrangement, the tubes are then soldered together and machined to provide the 25-degree half angle at the tip.

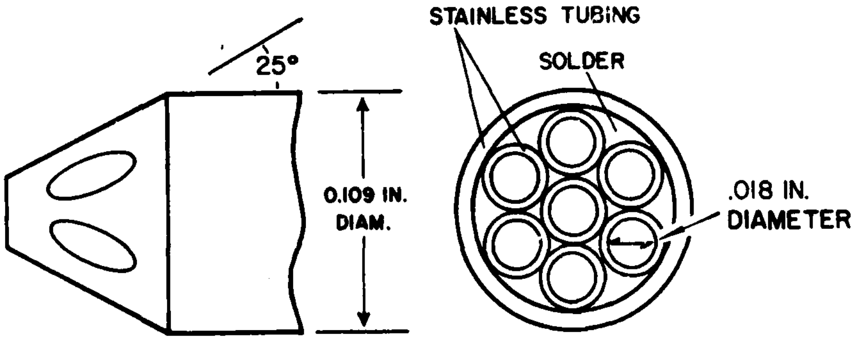


Figure 10. Probe Geometry

Probe Mounting Hardware

The test facility we used has a variable position sector capable of traversing 25 degrees in either direction for a total sweep of 50 degrees. Yet, for a complete calibration of the probe, a range of 0 through 80 degrees is necessary. Consequently, we constructed two stings (see Figure 11): a 15-degree bent sting for low angle measurements from -10 to 40 degrees angle of pitch, and a 55-degree bent sting for high angle measurements from 30 to 80 degrees angle of pitch. In addition, both stings were designed to permit the tip of the probe to pivot about a fixed point in the center of the tunnel.

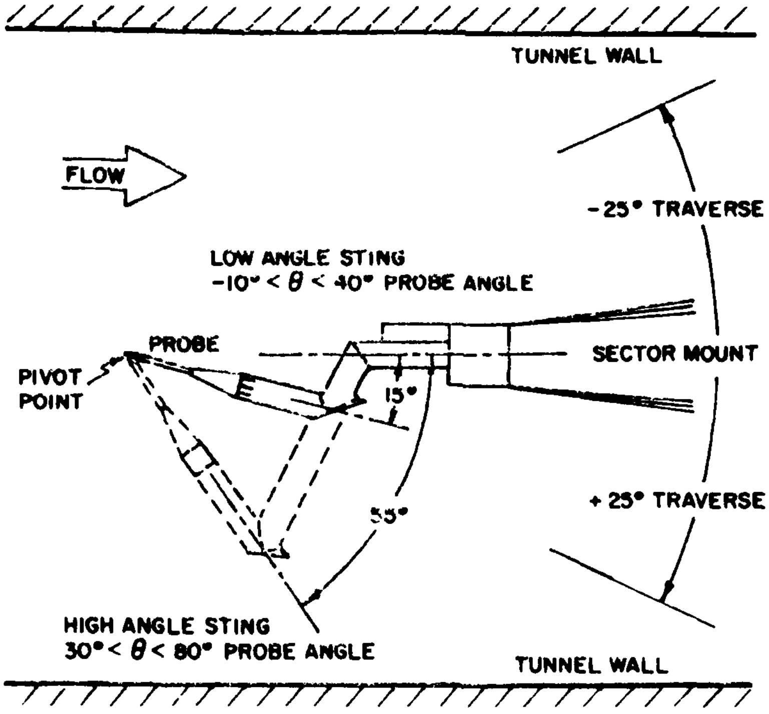


Figure 11. Sting Geometry

This insured uniform flow over the tip, despite changes in the angle of pitch.

Referring to Figure 12, each sting has 36 holes drilled in the front face. These holes are evenly spaced at 10-degree intervals and circumscribe a complete circle in roll. In this way, roll angle is accurately set by engaging the alignment pin on the probe holder with the alignment hole on the sting's face.

Data Point Selection

Since there are a total of 36 discrete roll angles, we are allowed to test six different roll angles in each outer sector. This is the primary reason why we chose the 6 x 6 Latin Square. Although the technique of Latin Squares allows us to conveniently sample large data sets, there are some drawbacks associated with seven-hole probe calibration. Primarily, we can no longer test the entire range of data and then allocate the data to a given sector based on the highest port pressure. Instead, we must determine beforehand which points will be tested and the sector they belong to. As a result, we are forced to draw the angular cutoffs for each sector without knowing where they actually lie.

In terms of angle of pitch, experience with incompressible calibrations has shown that 30 degrees is the smallest angle at which almost all points still fall in the outer

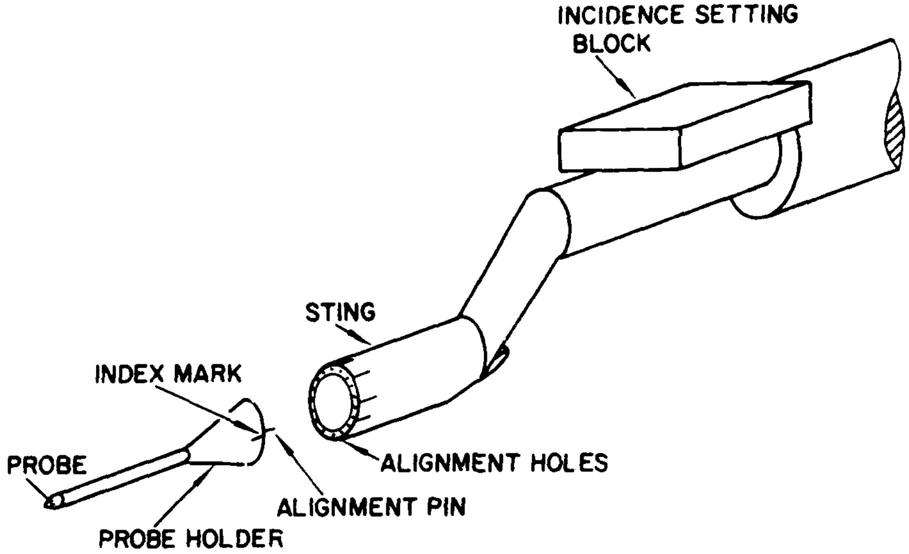
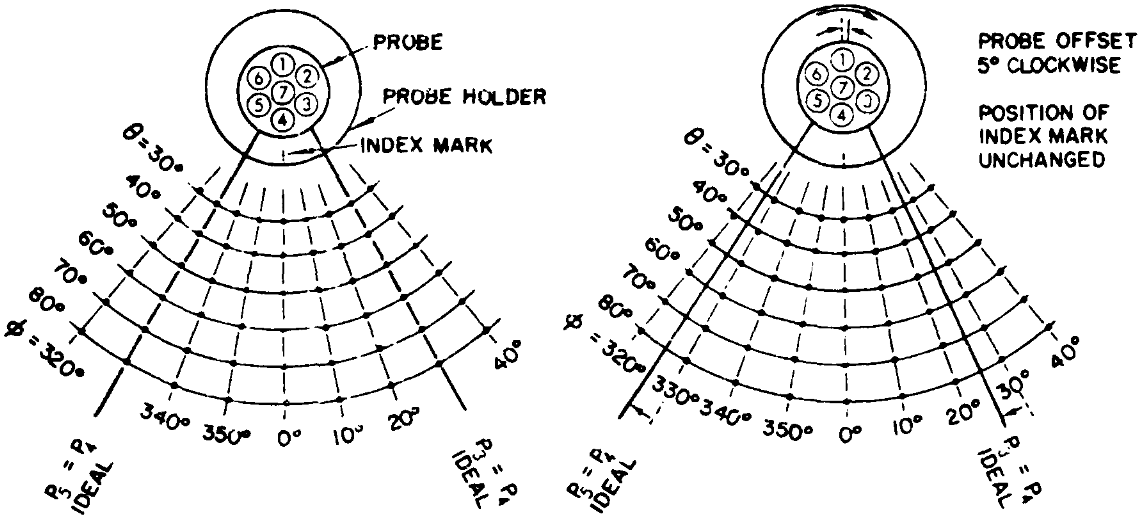


Figure 12. Probe, Holder, and Sting Assembly

sectors. Consequently, we selected 30 degrees as the lower limit of high angle measurement. And to match the number of roll measurements with an equal number of *pitch settings*, we measured pitch from 30 to 80 degrees in 10-degree increments for the outer sectors. Specifying the roll angles for each sector, however, presented some difficulties. Assuming a perfectly constructed probe, the outer sector boundaries, formed by the isobars between two adjacent peripheral ports, are coincident with a roll angle on which data is taken. This predicament is illustrated in Figure 13a for sector four. In this example, it is uncertain whether the points along the ϕ = 330 degrees line belong in sector four or five, or whether the points on the ϕ = 30 degrees line belong in sector three or four. To resolve this problem, we rotated the probe 5 degrees and then permanently fixed it to the holder in this position. From Figure 13b we see that the isobars have shifted off the points and that each sector has six clearly defined roll angles. Realistically, the probe is not constructed perfectly, causing the isobars to deviate from their ideal positions. But even so, the isobars can deviate up to 5 degrees in either direction before the points of one sector fall in another sector. At this point, one might ask if the probe can be accurately calibrated with the given offset. In actuality, *the offset makes no difference.* Since only the probe was rotated, leaving the index mark of the probe unchanged, the calibration process will interpret the pressures in terms of the roll angle set by the index mark. An alternative to the offset technique would use the roll angles between those presently measured (i.e., 5, 15, 25 . . . 355 degrees); however, this is an option we chose not to take.

Now that we've determined the angular coordinates of the data points to be taken in

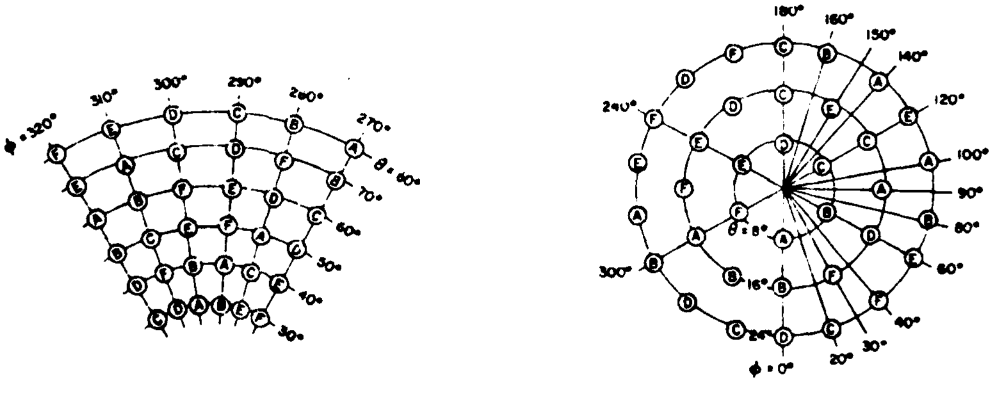


A. Probe and Holder Before 5-Degree Offset

B. Probe and Holder After 5-Degree Offset

Figure 13. Probe and Holder Before and After Offset for Sector Four

the outer sectors, it's time we considered the inner sector. Since we cannot evenly represent the inner sector with a square matrix in θ and ϕ, we simply covered the parameter space with an even density of 36 data points. The values of pitch angle were 8, 16, and 24 degrees, with roll angles selected to insure a uniform distribution. The 6 x 6 Latin Square matrix was then systematically filled with these ordered pairs. Although data were taken in terms of pitch and roll, these angles were converted to angles of attack and sideslip prior to calibration calculations.



A. Typical Outer Sector (Sector 5)

B. Inner Sector

Figure 14. Distribution of Data Points

Since the angular coordinates of all data points are known, the last task is to determine the Mach number at which each is sampled. The Mach numbers for each of the six letters in the Latin Square of Figure 9 are: A, M = 0.37; B, M = 0.45; C, M = 0.53; D, M = 0.66; E, M = 0.77; F, M = 0.91. The resulting distribution of data points, and the Mach number each was tested at, can be inferred from Figure 14, which depicts the inner sector and a typical outer sector.

Test Procedure and Software

The experiment was conducted in the 1 x 1 foot blowdown wind tunnel at the United States Air Force Academy. All pressure measurements were made with a Scanivalve Corporation Model T Scanivalve and pressure transducer, calibrated by a Wallace and Tiernan sonar manometer digital U-tube. Data were collected and reduced by a Digital PDP 11/45 computer and LPS-11 Laboratory Peripheral System.

Before any data were taken, the test conditions had to be set for each run. Angle of pitch was set first with a Gunner's quadrant. The quadrant was clamped to the sting's incidence setting block (refer to Figure 12) and set to the desired angle of pitch minus the angle of the elbow on the sting (either 15 or 55 degrees). Next, the sector was hydraulically positioned to level the quadrant. Since the sector normally has a tendency to drift between runs, we froze the mechanism by driving it against blocks. Shims were used to fine tune the adjustments. Using this technique, the pitch was set to within ±0.0112 degrees of the desired angle, and without drift. Once set, the pitch was not changed until all points at that angle were taken. The roll angle was set by pulling the spring-loaded probe forward to clear the alignment pin, then rotating it to the desired setting and engaging it into the new position. For a complete 360-degree revolution, the probe was first rotated counterclockwise (as viewed from the front) from 0 to 180 degrees in 10-degree increments. Next, it was rotated clockwise to the 190-degree position. And finally, the probe was rotated counterclockwise from 190 to 350 degrees in 10-degree increments. This subjected the tubes to a maximum twist of 180 degrees. After setting the roll angle, the tunnel was closed by a hydraulic ram and bolted shut. Mach number was then set by manually adjusting the tunnel's inner geometry with a series of hand cranks on the outside of the tunnel.

Prior to each run, the above variables were verified and input into the computer by a program titled TRISHP. Once the input conditions were set, air was blown, and after a few seconds to acquire steady-state conditions, data were recorded. Following this, the tunnel was reopened and the process repeated for a new data point. Typically, each run took about ten minutes to set up. Consequently, tests ran for over a week to acquire all 252 data points.

After all data were taken, a second program titled TRICAL performed the matrix operations, determined the calibration coefficients, and estimated the accuracy of the polynomial expansions in fitting the known data. Having done this, the probe calibration is complete.

V. Discussion

The primary purpose of this experiment is to create a power series curve fit which accurately determines the actual flow conditions from pressures measured on the probe. The best way to evaluate the performance of the curve fit is to analyze the standard deviations between the experimental data and the polynomial determination of those data. These standard deviations are presented in Table 1. In addition, the standard deviations obtained from past incompressible calibrations are included in Table 2 for comparison (Ref. 1).

From Table 1, the standard deviations of α_T and β_T are 0.78 degrees and 0.72 degrees respectively. Although these values appear to be quite good, they are significantly higher than those obtained from the incompressible calibrations. Specifically, the incompressible flow calibrations produced standard deviations of 0.42 degrees for α_T and 0.36 degrees for β_T. Similar variations are also noted in the percent errors of total and dynamic pressures. Yet, when we recall that the incompressible calibrations used a fourth order power series with 80 data points per sector and the compressible calibration used a third order power series with only 36 data points per sector, the slight differences are quite understandable. Still another measure of curve-fit accuracy

Table 1

STANDARD DEVIATIONS
COMPRESSIBLE FLOW CALIBRATION

INNER SECTOR		OUTER SECTORS	
SECTOR / EXPRESSION	7	SECTOR / EXPRESSION	AVERAGE 1 - 6
$\sigma(\alpha_T)$	0.78°	$\sigma(\theta)$	4.27°
$\sigma(\beta_T)$	0.72°	$\sigma(\phi)$	0.57°
$\dfrac{\sigma(P_{oL})}{P_{oL} - P_{\infty L}}$	2.5%	$\dfrac{\sigma(P_{oL})}{P_{oL} - P_{\infty L}}$	5.7%
$\dfrac{\sigma(P_{oL} - P_{\infty L})}{P_{oL} - P_{\infty L}}$	1.4%	$\dfrac{\sigma(P_{oL} - P_{\infty L})}{P_{oL} - P_{\infty L}}$	2.2%
$\sigma(M)$	0.006	$\sigma(M)$	0.061

Table 2

STANDARD DEVIATIONS
INCOMPRESSIBLE FLOW CALIBRATIONS (Ref. 1)

Average of Inner Sector		Average of 6 Outer Sectors	
EXPRESSION	Std. Dev.	EXPRESSION	Std. Dev.
$\sigma(\alpha_T)$	0.42°	$\sigma(\theta)$	0.84°
$\sigma(\beta_T)$	0.36°	$\sigma(\phi)$	1.17°
$\frac{\sigma(P_{oL})}{P_{oL} - P_{\infty L}}$	0.6%	$\frac{\sigma(P_{oL})}{P_{oL} - P_{\infty L}}$	1.2%
$\frac{\sigma(P_{oL} - P_{\infty L})}{P_{oL} - P_{\infty L}}$	1.0%	$\frac{\sigma(P_{oL} - P_{\infty L})}{P_{oL} - P_{\infty L}}$	2.4%

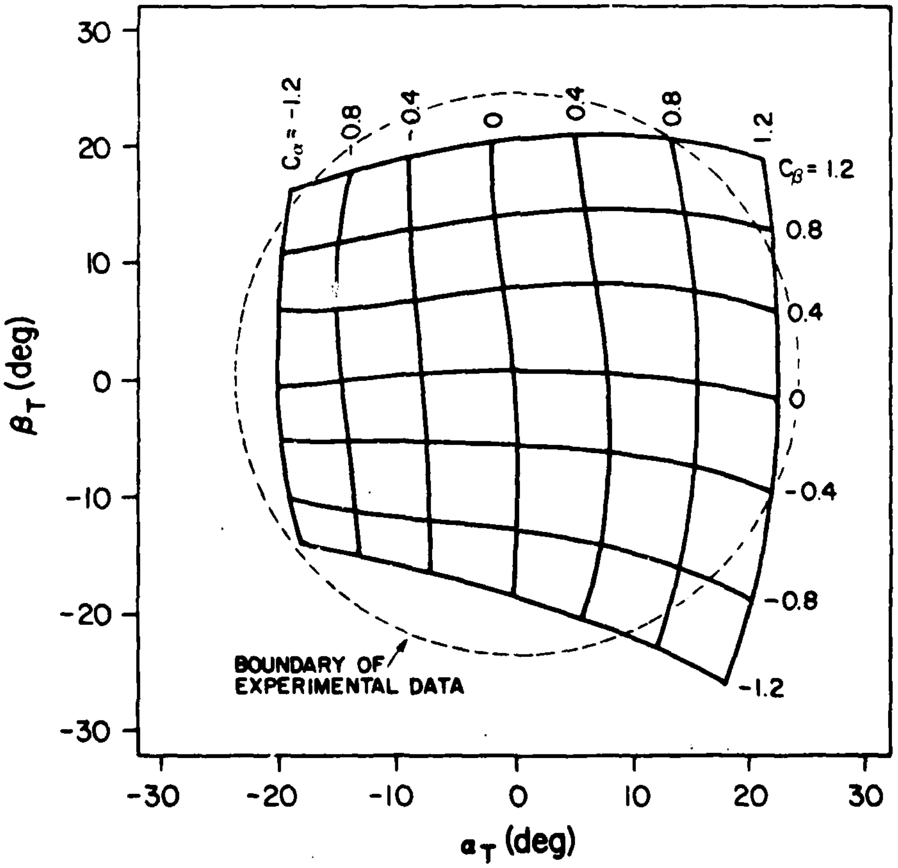


Figure 15. Isolines of C_α and C_β versus α_T and β_T for Low Angles at Low Mach Number

in the inner sector is apparent from Figures 15 and 16. Both of these figures depict isolines of C_α and C_β plotted against α_T and β_T with C_M held constant. Figure 15 presents this information for a low Mach number, with Figure 16 representative of a high Mach number. In both cases, the isolines of the coefficients are nearly straight and relatively orthogonal to each other, implying a linear dependence on their respective angle and independence to the other angle. These properties, however, begin to break

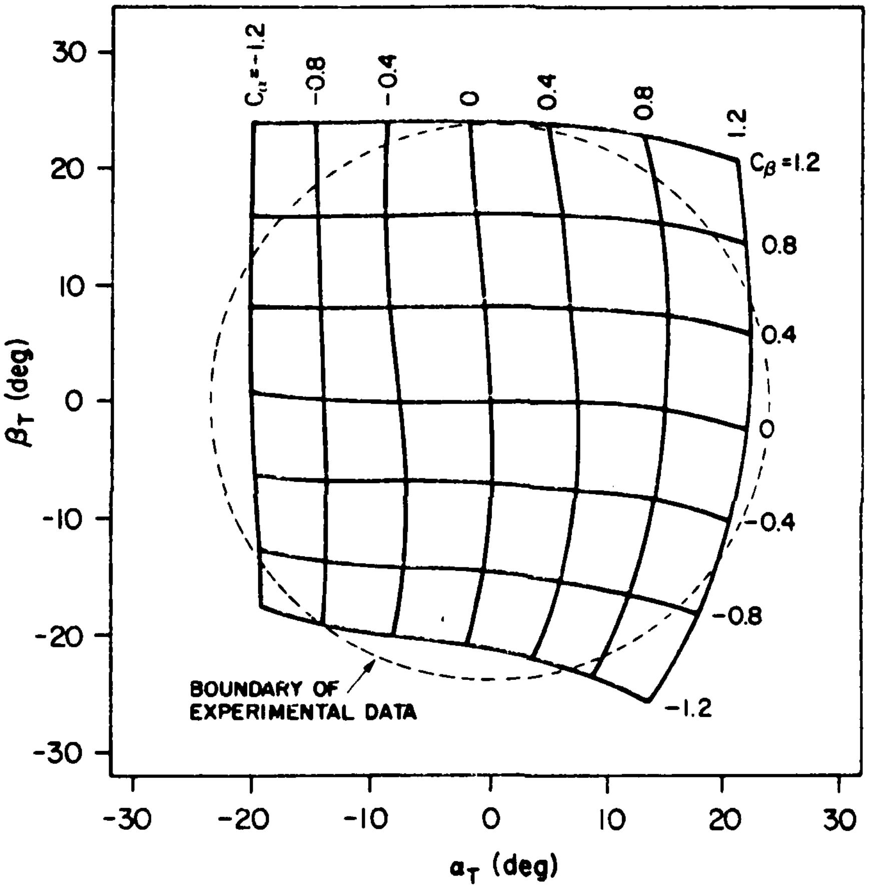


Figure 16. Isolines of C_α and C_β versus α_T and β_T for Low Angles at High Mach Number

down as we exceed the limits of our experimental data. But since the boundaries of the experimental data coincide with the inner-outer sector interface (see Figure 17), data points lying outside the inner sector's experimental boundary will fall into the outer sectors. Consequently, tne accuracy in determining flow properties is uniform throughout the inner sector.

In the outer sectors, the average standard deviation of the error in calculating ϕ is 0.57 degrees for the compressible calibration and 1.17 aegrees for previous incompres-

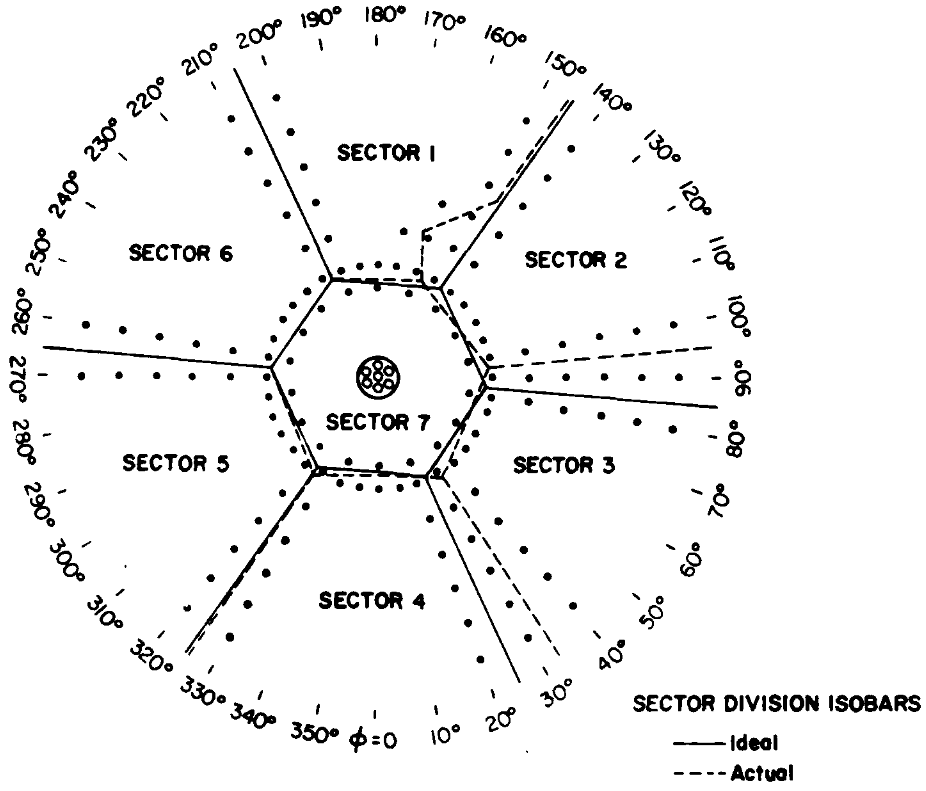


Figure 17. Ideal and Actual Sector Boundaries Based on the Highest Probe Measured Pressures

sible calibrations. Judging from Figure 17, one might expect the average standard deviation of ϕ for the compressible calibration to be substantially greater than that for the incompressible calibrations, because the actual boundaries of sectors one through four do not coincide with the ideal boundaries for which the calibration was made. Although this phenomenon is entirely consistent with the manufacturing anomalies associated with probe construction, the actual boundaries cannot be determined in advance and, therefore, cannot be taken into account in a calibration scheme using the method of Latin Squares. Nevertheless, despite the actual locations of the sector boundaries, the standard deviation in ϕ for the compressible calibration agrees favorably with its incompressible counterpart. The reason for this is shown in Figure 18. Even though Figure 18 is based on data taken from incompressible calibrations, it is representative of seven-hole probes in general. That is, the coefficient of roll continues to behave linearly, or in a way that can be represented easily by a polynomial, slightly beyond the isobaric sector boundaries. As such, the effect of using data within 10 degrees in roll from the selected boundaries, as Figure 18 suggests, does not have an adverse effect on fitting the data. Consequently, the disagreement between the actual and ideal sector boundaries

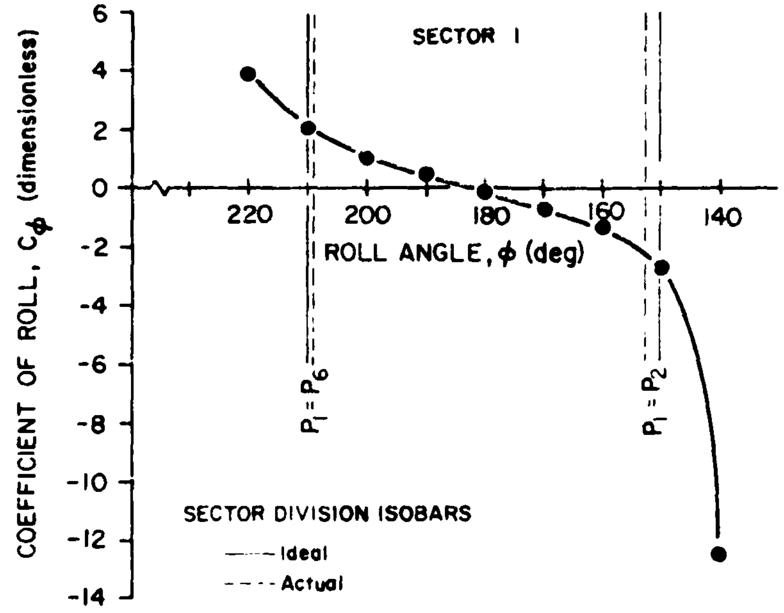


Figure 18. Incompressible Data Showing C_ϕ versus : in Sector 1 for ε = 84 Degrees

of Figure 17 does not signifi.:antly influence the determination of roll angle.

Despite the small error in calculating the roll angle, the standard deviation of the error in calculating pitch angle at 4.27 degrees is unacceptably high, especially when compared to the 0.84-degree standard error of previous incompressible calibrations. The actual variation between the experimental data and its polynomial prediction is presented in Figure 19, which depicts a typical outer sector. From this illustration, we can see that the greatest error in determining the pitch angle occurs at high angles of attack. This occurs as a result of the polynomial's inability to fit the actual data. Typically, the shape of the C_θ versus θ curve looks much like the lift curve of a stalling airfoil. Figure 20 illustrates such a curve for sector one, based on the extensive data available from incompressible probe calibrations. The reason why the curve hooks over as it does is evident after examining the two pressures comprising the numerator of the coefficient of pitch. According to Figure 20, the center port pressure decreases with increasing angle of pitch. Beyond some point, a suction develops at this port, causing the pressure there to dip below the free stream static pressure. But near 80 degrees angle of pitch, the suction breaks and the pressure begins to increase. As this occurs, the slope of the Cp_7 curve approaches the slope of the Cp_1 curve. Once the two slopes are equal, the rate of change of the numerator is zero, causing the slope of the coefficient of pitch also to be zero. No calibration may be made beyond this point, because each value of C_θ then corresponds to two values of θ. And since θ is a function of C_θ, the poly-

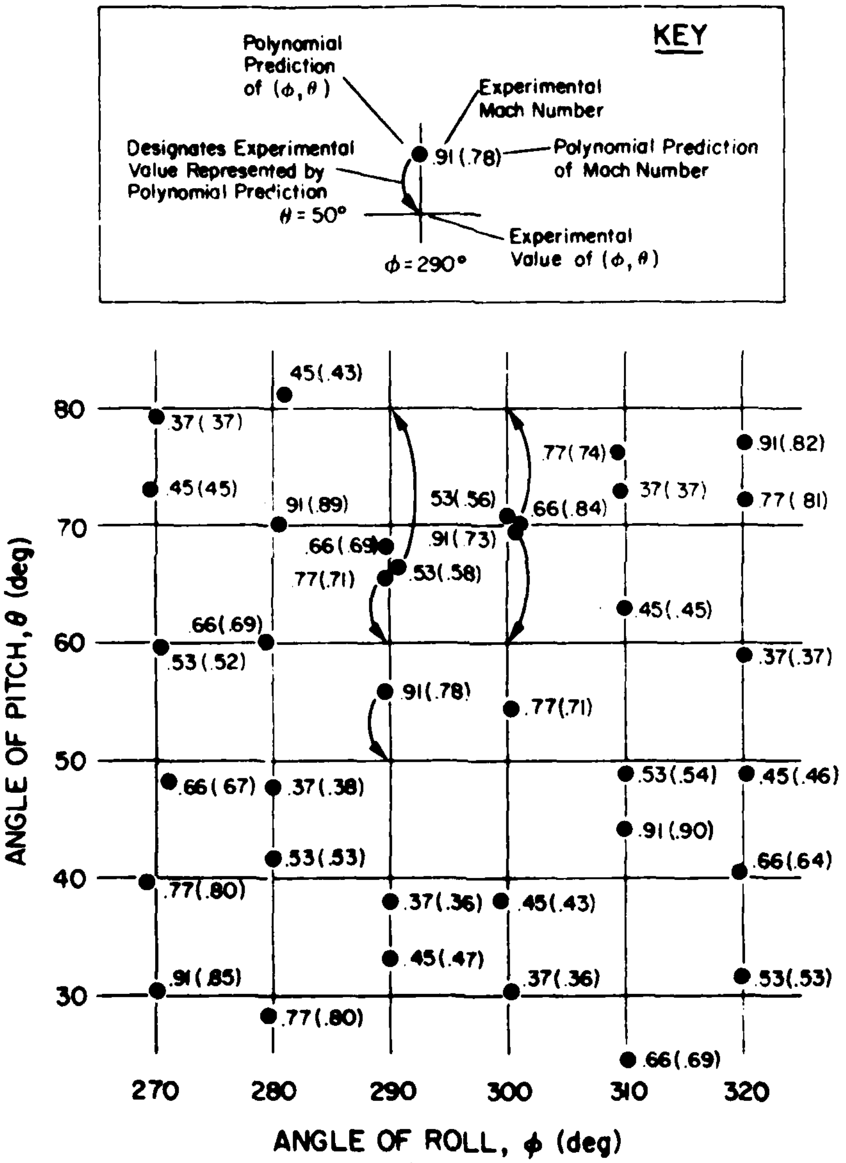


Figure 19. Three-Dimensional Data Set Showing Correlation Between Experimental and Polynomial Data for Sector 5

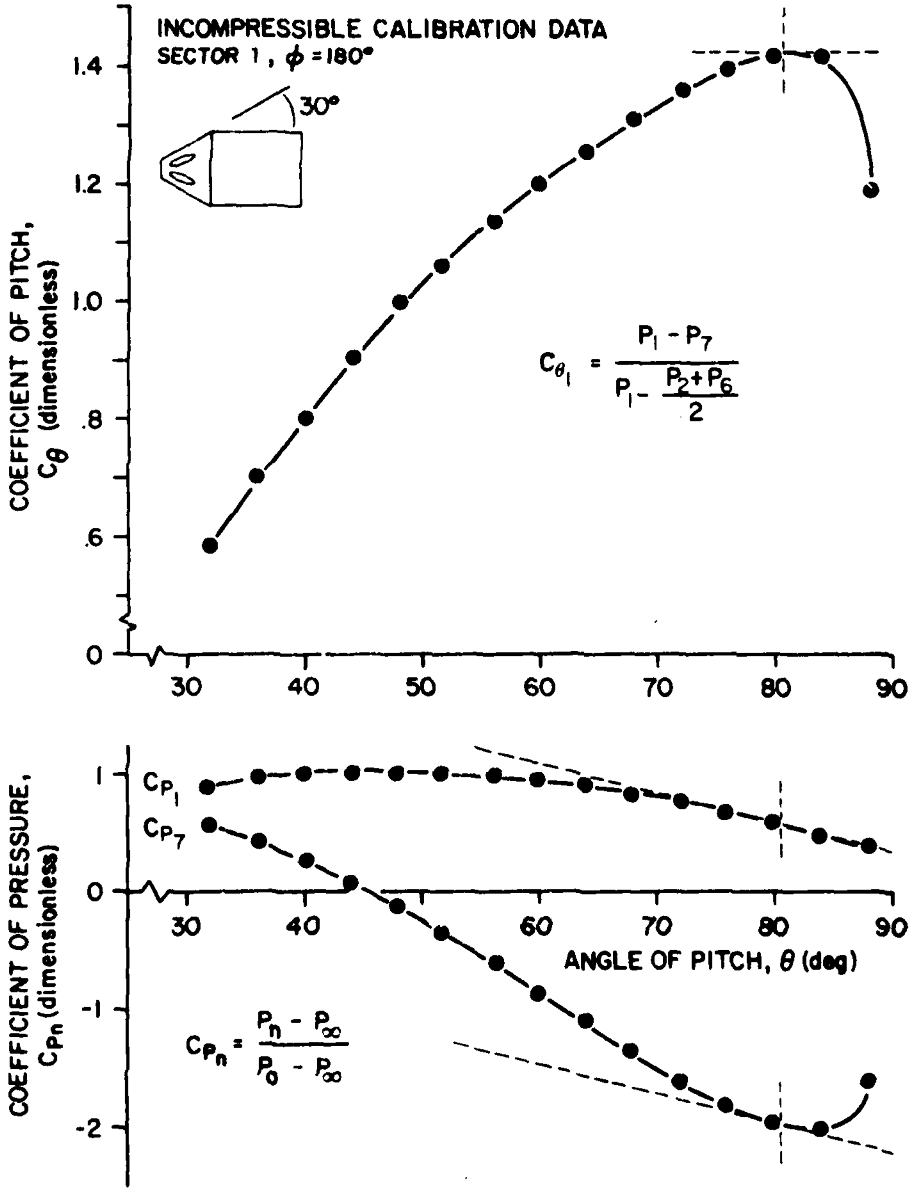


Figure 20. Mechanism for the Breakdown of Linearity in the Coefficient of Pitch at High Angles of Attack

nomial can only calculate one unique value of θ for a given value of C_θ. Therefore, based on the information presented in Figure 20, we would expect to be capable of calibrating the probe out to 80 degrees angle of pitch. This was indeed the case for the past incompressible calibrations, but not the case in our compressible calibration, as Figure 19 vividly points out.

The failure of the calibration at high angles of pitch is most likely the result of probe geometry. Specifically, the half angle of the probe used in our compressible calibration is 25 degrees (see Figure 10) as opposed to the 30-degree half angle of the probes used in the incompressible calibrations. This steeper half angle was incorporated to permit a closer approximation of total pressure (as measured by the peripheral ports) at high angles, in the hope of extending the range of calibration beyond 80 degrees of pitch. However, this reasoning overlooked the effect increasing the half angle would have on the central port. That is, a steeper half angle requires the flow passing around the probe tip to turn through a greater angle, causing the suction over the central port to break at a lower angle of pitch. Consequently, the slope of the coefficient of pitch levels off earlier, limiting the calibration to a pitch angle below 80 degrees. But even as this theoretical upper limit of the calibration is approached, the breakdown in linearity of the coefficient of pitch with increasing angle of pitch forces the polynomial to work harder to fit the data. Since at high angles of attack small changes in the coefficient of pitch result in large changes in the angle of pitch, a small mismatch between the actual data and the polynomial curve fit translates into a large error as Figure 21 illustrates. Thus, as a result of the difficulty in fitting the actual data near the theoretical limit of calibration as established by the probe's geometry, the calibration appears to be limited to an angle of pitch below 80 degrees.

To confirm these suspicions, we ran a second data reduction, excluding all 80-degree pitch data. The standard deviations of the truncated data set are displayed in Table 3

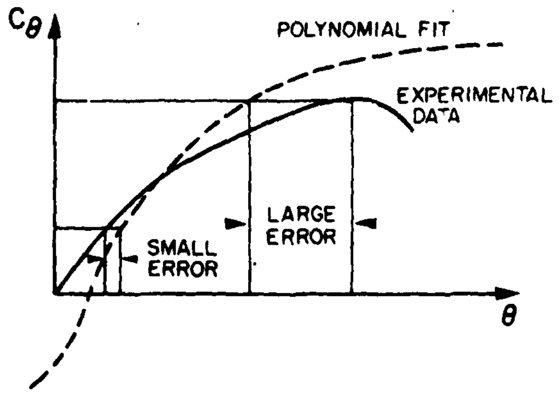


Figure 21. Limitations of Third Order Curve Fit at High Angles of Attack

Table 3

STANDARD DEVIATIONS
COMPRESSIBLE FLOW CALIBRATION

INNER SECTOR		OUTER SECTORS (θ = 80° Data Truncated)	
SECTOR EXPRESSION	7	SECTOR EXPRESSION	AVERAGE 1 - 6
$\sigma(\alpha_T)$	0.78°	$\sigma(\theta)$	0.79°
$\sigma(\beta_T)$	0.72°	$\sigma(\phi)$	0.39°
$\dfrac{\sigma(P_{oL})}{P_{oL} - P_{\infty L}}$	2.5%	$\dfrac{\sigma(P_{oL})}{P_{oL} - P_{\infty L}}$	1.1%
$\dfrac{\sigma(P_{oL} - P_{\infty L})}{P_{oL} - P_{\infty L}}$	1.4%	$\dfrac{\sigma(P_{oL} - P_{\infty L})}{P_{oL} - P_{\infty L}}$	4.1%
$\sigma(M)$	0.006	$\sigma(M)$	0.022

and reflect a significant decrease in the error associated in calculating the experimental quantities. For example, the standard deviation in calculating pitch angle went down from 4.27 degrees to 0.79 degrees. The correlation between the data points of sector five also improved dramatically and is depicted in Figure 22. However, one must keep in mind that since the 80-degree points were removed from the data set, the data set is no longer as accurately represented. That is, in addition to losing the 80-degree data, we also lost an equal number of associated roll angle and Mach number data. Nevertheless, the greatly reduced standard deviations support our contention that for our probe in the vicinity of 80 degrees, the third order curve fit is incapable of accurately fitting the data.

VI. Conclusions and Recommendations

The polynomial expansion in three variables accurately extends the calibration of seven-hole probes into the compressible regime. Based on the reasonably close correlations between the standard deviations of past incompressible calibrations and the standard deviations of the compressible calibrations, the method of Latin Squares furnishes a sample space which accurately represents a much larger three-dimensional parameter space. Finally, the third order curve fit accurately represented the parameter space

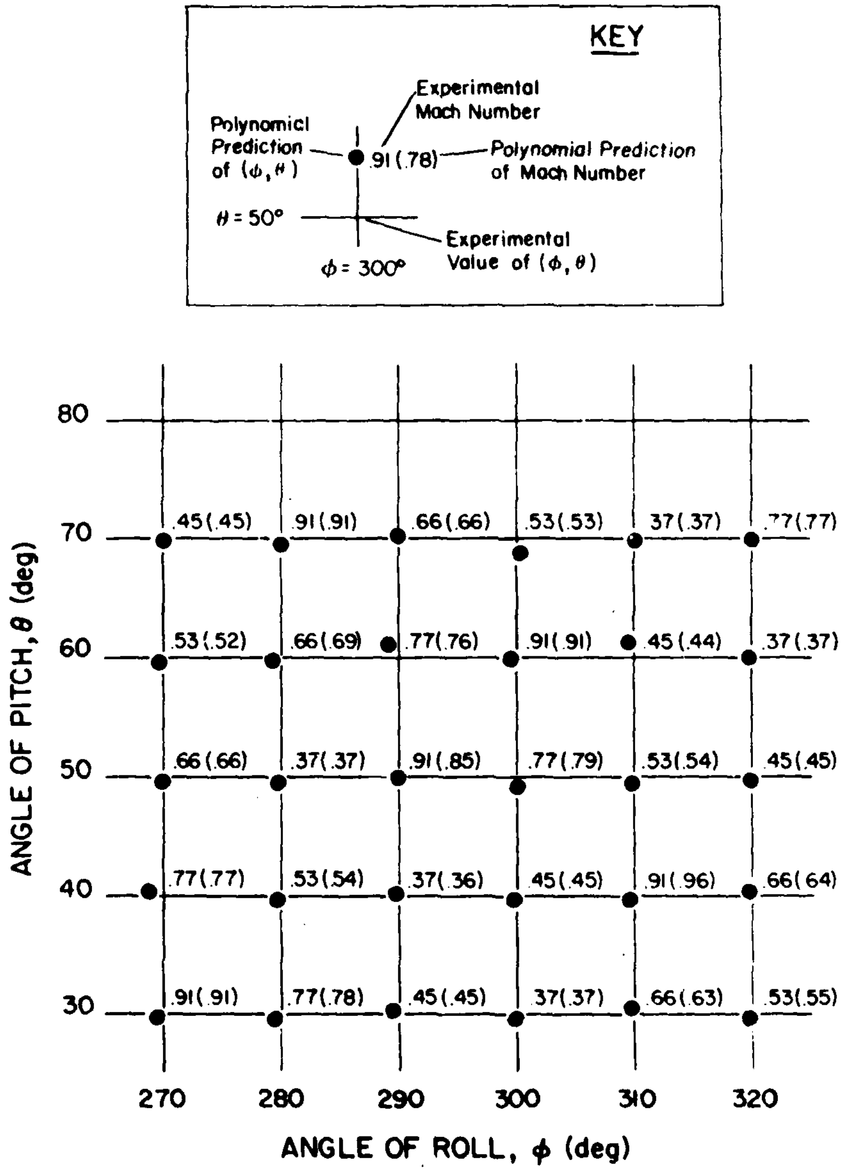


Figure 22. Three-Dimensional Data Set Showing Correlation Between Experimental and Polynomial Data for the Truncated Data Set of Sector 5

out to an angle of pitch of 70 degrees, but fell completely apart when required to fit the data extending to 80 degrees angle of attack.

It is recommended that an additional Latin Square be run in an outside quadrant. A set of calibration coefficients can then be determined using that Latin Square of data only; the same should be done with the original square. The results can then be examined to see how well the calibration coefficients of one square determine the values of the other square and vice versa. Close correlations should establish beyond a reasonable doubt the validity of a Latin Square in representing a parameter space. Next, both Latin Squares should be combined to determine another set of calibration coefficients, which in turn can be used to see if more data points significantly improve the degree of fit. Lastly, using the combined set of data points, extend the calibration to a fourth order polynomial and examine it to see if the addition of the fourth order terms significantly improve the degree of fit up to 80 degrees angle of attack.

VII. Acknowledgements

The authors of this paper wish to acknowledge the assistance of several individuals, who without their help and expertise, this calibration would not have been possible. In addition to the various technicians who installed the equipment and operated the tunnel, Mr. Claude Hollenbaugh was chiefly responsible for constructing the seven-hole probe as well as the holding and indexing apparatus. Capt. Tom Bolick wrote the computer software which performed the data acquisition and reduction for the probe calibration. And finally, Lt. Col. Roger Gallington, who engineered the theory of seven-hole probes, provided the background on incompressible flow calibrations and furnished guidance for the extension to compressible flow.

Symbols

A_i		the ith value of a particular data point, where A is either α_T, β_T, C_o, or C_q for low angles and θ, ϕ, C_{o_n}, or C_{q_n} for high angles
A_{EXP_i}		the ith value of the experimentally known value of A
A_{POLY_i}		the ith value of the polynomial predicted value of A
C_{M_n}	n=1-7	coefficient representative of compressibility effects
C_o		apparent total pressure coefficient for low angles
C_{o_n}	n=1-6	apparent total pressure coefficient for high angles
C_q		apparent dynamic pressure coefficients for low angles
$\overline{C}_{q_n}$	n=1-7	average value of the apparent dynamic pressure coefficients for a given sector
C_{q_n}	n=1-6	apparent dynamic pressure coefficients for high angles

C_{p_n}	n=1-7	coefficient of pressure (arrived at by nondimensionalizing an individual port pressure)
C_α		angle of attack pressure coefficient for low angles
C_{α_n}	n=1-3	intermediate pressure coefficients used to determine C_α and C_β
C_β		angle of sideslip pressure coefficient for low angles
C_{θ_n}	n=1-6	pitch angle pressure coefficient for high angles
C_{ϕ_n}	n=1-6	roll angle pressure coefficient for high angles
K_n^A	n=1-20	calibration coefficient, where A denotes the parameter to which a particular set of K's belong, and n denotes the particular term of the calibration coefficient in the polynomial power series expansion
M		Mach number of free stream
M_L		local Mach number
P_n	n=1-7	pressure at port "n"
P_o		total pressure of free stream
P_{oL}		local total pressure
P_∞		static pressure of free stream
$P_{\infty L}$		local static pressure
$\overline{P}_{1-6}$		average of pressures 1 through 6
u,v,w		local velocity components with respect to probe
V		local velocity with respect to probe
α		angle of attack
α_T		angle between probe axis and velocity vector projected on vertical plane through probe's x-axis
β		angle of sideslip
β_T		angle between probe axis and velocity vector projected on horizontal plane through probe's x-axis
θ		total angle between velocity vector and probe's x-axis
γ		ratio of specific heats
ϕ		angle between the plane containing the velocity vector and the probe's x-axis and the x-z plane measured positive clockwise from port number four as viewed from the front
σ		standard deviation

References

1. Gallington, R. W. "Measurement of Very Large Flow Angles with Non-Nulling Seven-Hole Probes." Aeronautics Digest - Spring/Summer 1980, USAFA-TR-80-17, USAF Academy, Colorado 80840, pp. 60-88.

2. Robertson, J. A. and Crowe, C. T. "Wave Propagation in Compressible Fluids." Engineering Fluid Mechanics, 1st ed. Boston: Houghton Mifflin Co., 1975, p. 370.

3. Chow, C. Y. and Kuethe, A. M. "Flow Around Spheres and Circular Cylinders." Foundations of Aerodynamics: Bases of Aerodynamic Design, 3rd ed. New York: John Wiley and Sons, 1976, p. 375.

4. Netter, J. and Wasserman, W. "Simple Linear Regression Model in Matrix Terms." Applied Linear Statistical Models. Illinois: Richard D. Irwin, Inc., 1975, p. 200.

5. Barker, K. W., Gallington, R. W., and Minster, S. N. "Calibration of Five-Hole Probes for On-Line Data Reduction." Aeronautics Digest - Spring 1979, USAFA-TR-79-7, USAF Academy, Colorado 80840.

6. Cochran, W. G. and Cox, G. M. "Double Grouping: Latin Squares." Experimental Designs, 2nd ed. New York: John Wiley and Sons, 1957.

7. Holman, J. P. "Statistical Analysis of Experimental Data." Experimental Methods for Engineers, 3rd ed. New York: McGraw-Hill, 1978, p. 51.

8. Kline, S. J. and McClintock, F. A. "Describing Uncertainties in Single-Sample Experiments." Mechanical Engineering, January 1953.

SEVEN-HOLE PROBE DATA ACQUISITION SYSTEM

G. Sisson*, T. Bolick**, F. Jonas**

Abstract

This paper describes the computer software (programs) used in conjunction with the seven-hole probe. The complete data acquisition system includes the seven-hole probe, a computer-driven traverse mechanism, computerized data acquisition equipment, and computer graphic displays.

I. Introduction

The USAF Academy and NASA-Ames Research Center have conducted a joint research effort over the past three years which has resulted in a new means of surveying aerodynamic flow fields. A newly invented seven-hole probe was used to measure pressures and velocities in the wake of wind tunnel models as part of this effort (Ref. 1-3). With this flow field information the aircraft designer can improve aircraft designs, increasing range performance, handling characteristics, and overall mission performance.

Flow field or wake surveys are not new. They have been performed for years with different types of instrumentation such as hot-wire anemometers, pitot tubes, four- and five-hole probes, and laser doppler velocimeters and anemometers. Each type of instrumentation has inherent advantages and disadvantages. The seven-hole probe is a new form of instrumentation that capitalizes on high data rates, ease of use, low costs, and an excellent range of applicability.

The first seven-hole probe surveys (Ref. 1) resulted in the measurement of local total, static, and dynamic pressures to within 5 percent accuracy (95 percent confidence level) and local flow angles to within ±2 degrees (95 percent confidence level) for incompressible flows.

*Hewlett-Packard, Ft. Collins, Colorado
**Captain, USAF, Department of Aeronautics, DFAN

These results were very impressive. Nevertheless, certain improvements were possible.

The extension of the probe calibration to compressible subsonic flows (Ref. 3) was one such improvement. This is very significant for two reasons. One, most aircraft spend the majority of their flight time in this regime; and two, even in an incompressible flow field, wakes can be produced which exhibit compressibility.

The computer programs used in the survey technique (Ref. 2) placed many limitations on the user; thus another area for improvement was identified. The final data was not available while the wind tunnel was still running, making data validation difficult. The data was always collected at uniformly spaced points in the flow, which resulted in abnormally large run times or unusually sparce data sets. Thus, real time or on-line data analysis was not possible, resulting in longer test programs to identify pertinent flow field phenomena. In addition, the computer inputs were very cryptic and the programs were not reusable due to poor programming techniques and lack of program documentation.

This report discusses improvements to these computer programs that alleviate these limitations. The complete program documentation follows in the appendices for the seven-hole probe data acquisition system.

II. Discussion

Data validation was a problem with the old software. Four computer programs were executed in sequence to generate a plot which would tell the user if all the electrical and pneumatic connections for the experiment were correct. A schematic of this procedure is shown in Figure 1. The second program, TOPWING, was the only one that used the wind tunnel

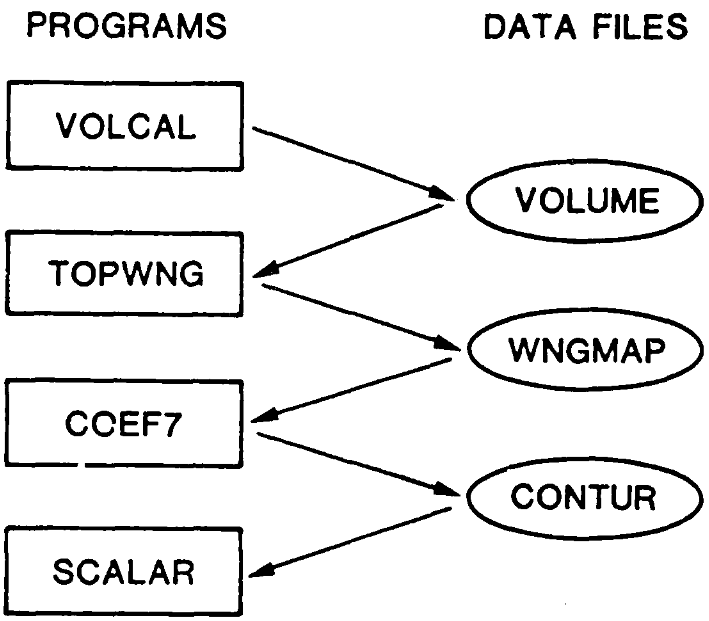


Figure 1. Schematic of Old Software

apparatus. Because wind tunnel schedules are always tight, it was very tempting to run this program over and over until all necessary wind tunnel data was collected. Only then could the data be reduced, plotted, and analyzed. On more than one occasion the data proved to be erroneous or invalid because of changes in the electrical and pneumatic connections during the test.

The solution to this problem was to make a single computer program perform the same functions of the previous four programs. This single program could then be made to include data reduction and display while the test was being conducted, thus allowing on-line data analysis. Any data anomalies could then be detected immediately, allowing the user either to retake the data or inspect the instrumentation. However, a single program proved to be impossible, as it required too much space in the computer (RAM or core memory).

The new software consists of three programs that use a common data file format. These three programs are linked together through system level control language command files, thus abbreviating core storage requirements. The result is that the user sees only one task instead of

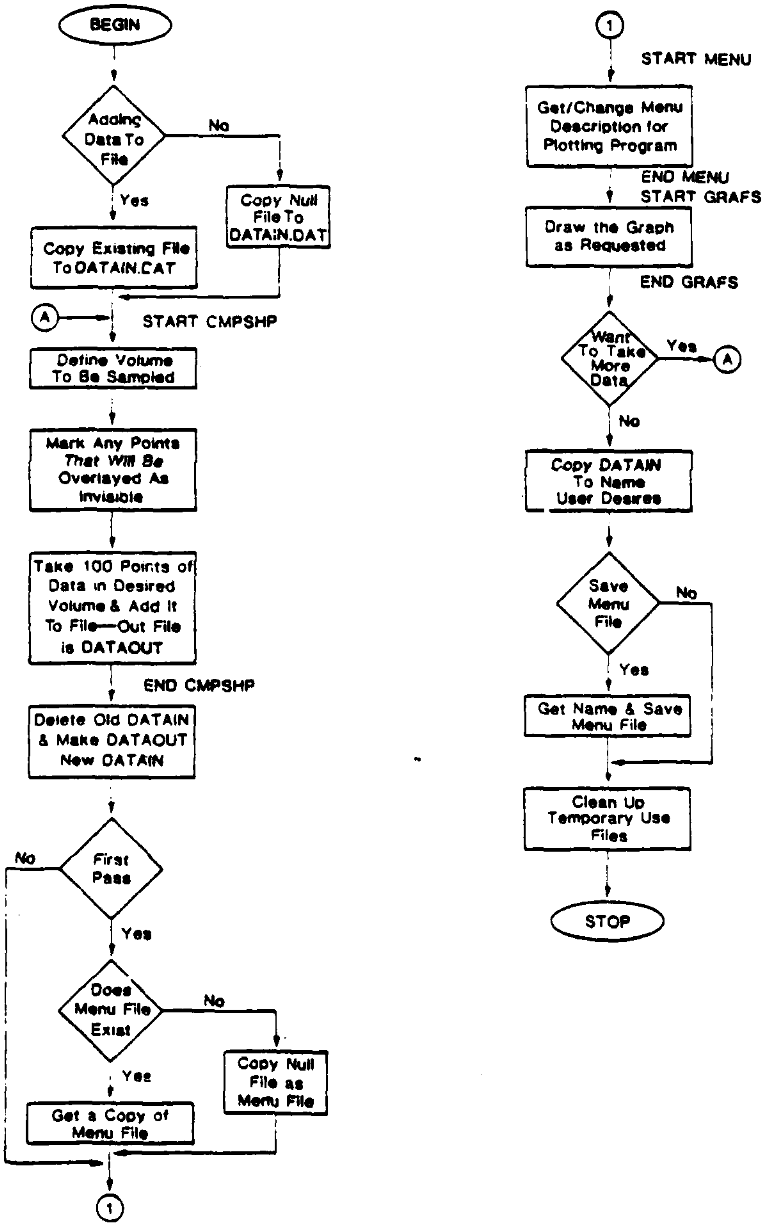

BEGIN
Adding Data To File
No
Yes
Copy Null File To DATAIN.DAT
Copy Existing File To DATAIN.CAT
A
START CMPSHP
Define Volume To Be Sampled
Mark Any Points That Will Be Overlayed As Invisible
Take 100 Points of Data in Desired Volume & Add It To File—Out File is DATAOUT
END CMPSHP
Delete Old DATAIN & Make DATAOUT New DATAIN
First Pass
No
Yes
Does Menu File Exist
No
Yes
Copy Null File as Menu File
Get a Copy of Menu File
1
1
START MENU
Get/Change Menu Description for Plotting Program
END MENU
START GRAFS
Draw the Graph as Requested
END GRAFS
Want To Take More Data
Yes
A
No
Copy DATAIN To Name User Desires
Save Menu File
No
Yes
Get Name & Save Menu File
Clean Up Temporary Use Files
STOP

Figure 2. Compressible Seven-Hole Probe DAS Schematic

several independent tasks. This is illustrated in Figures 2 and 3, which show the flow for data acquisition tasks and for the separate plotting

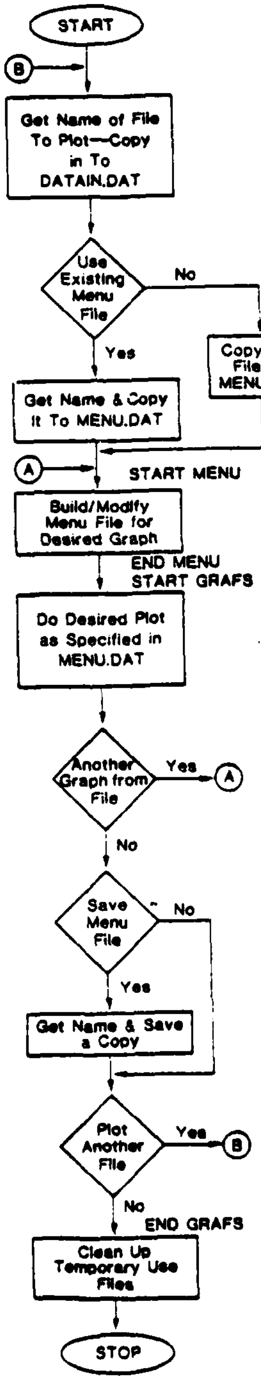

START
B
Get Name of File To Plot—Copy in To DATAIN.DAT
Use Existing Menu File
No
Yes
Copy Null File To MENU.DAT
Get Name & Copy It To MENU.DAT
A
START MENU
Build/Modify Menu File for Desired Graph
END MENU
START GRAFS
Do Desired Plot as Specified in MENU.DAT
Another Graph from File
Yes
A
No
Save Menu File
No
Yes
Get Name & Save a Copy
Plot Another File
Yes
B
No
END GRAFS
Clean Up Temporary Use Files
STOP

Figure 3. P?ots Program Schematic

tasks respectively. The complete program listings are in Appendix A. The first program, CMPSHP, performs the functions previously done by VOLCAL, TOPWING, and COEF7. It allows the user to specify the volume of space to be sampled using a three-degree-of-freedom traversing mechanism (Ref. 2). The data is then acquired in the defined space and reduced to pressure coefficients, angles, and velocity ratios. The second program, MENU, is a highly interactive program, which allows the user to define the desired graphic display of the data. The third program, GRAFS, uses the description provided by MENU and produces the desired graphic display of the data. The data can be displayed as contour and/or axonometric plots of the scalar data, and/or vector plots of the cross-flow velocity data. These last two programs, GRAFS and MENU, perform the same logial function as the old SCALAR program, but with an enhanced user interface and greater flexibility.

Using the old software, the seven-hole probe was positioned at points in a rectangular grid which had nearly uniform spacing between points, as shown in Figure 4. This was the most efficient spacing without a priori knowledge of the location of the pressure and velocity gradients in the flow. Nevertheless, it meant collecting a large amount of data in small grids for both the uniform flow and the gradient regions. This condition

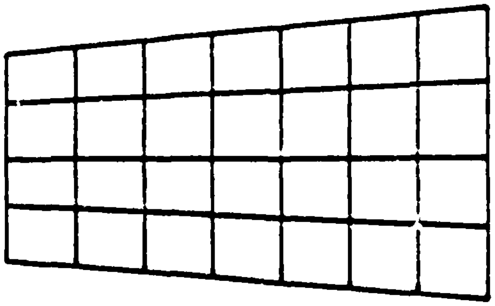


Figure 4. Uniformly Spaced Grid

was far from optimum from a data collection point of view. What was required was a non-uniformly spaced grid which concentrated points in regions of interest, as shown in Figure 5. Thus, the software was modified to display this non-uniformly spaced data as a series of grids. In addition, these grids were allowed to overlap so that two or more might cover the same position in the plots. This allowed more flexibility in collecting the data and still did not require an a priori knowledge of the location of major flow field features. Data acquisition is thus accomplished by first examining a widely-spaced grid. After examining the data, a newer, denser grid is then specified by the user to cover only those regions where more detail is required. This procedure can be repeated as many times as is required to get the necessary resolution. This is what is meant by interactive data collection.

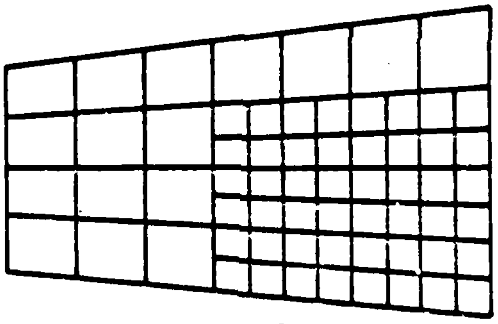


Figure 5. Non-Uniformly Spaced Grid

Although this technique solved the problem of non-uniformly spaced data, it caused another: which data should be used at a point where more than one grid is defined there? The obvious and perhaps best solution would be to average the data. The new software, however, does not perform any averaging in the interest of saving time and space. Instead, the points in the previously defined grids are marked as invisible by the newly defined grid, as shown in Figure 6, and illustrated in Appendix B.

With the previously discussed modifications, the new software provided for interactive data collection and plots of the data available within seconds after the data collection was completed. It was a simple, inexpensive, and highly accurate means of flow visualization, which provided quantitative results for many aerodynamic flow proper ,. The software still suffered one major flaw, however, in that it was not user-oriented. The keyboard input section of the display program was a series of questions. Running this program frequently required answering these same questions many times. This created a situation for possible errors and turned out to be a source of frustration for the user. Improving the input section of the display program was solved by moving all the question and answer software to the MENU program. The user responses were then written out to a scratch file, which was read in when the MENU program

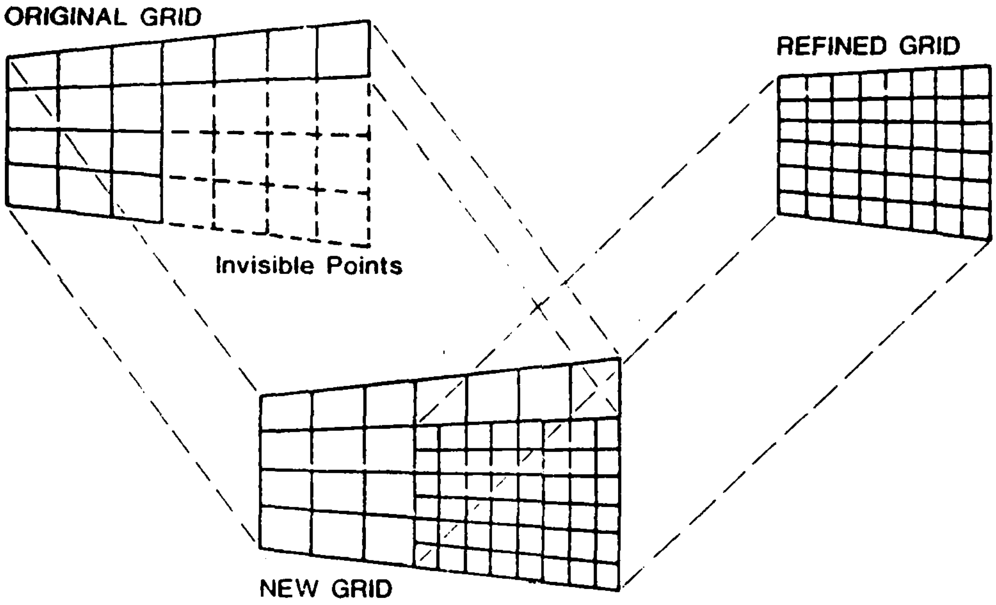


Figure 6. Multiple Grid Display with Invisible Points

was next executed. This data is then displayed on the computer terminal screen, and the user is free to change any parameter in any sequence. When the parameters displayed are what the user wants, a null input (no change desired) signals the program to plot the data. If the plot required is to be the same as the last one, then only a null input is required. This is much faster and less error-prone than the old approach. In addition, this allows the user to check his input parameters constantly. Thus, moving all the user input software to the MENU program has the added benefit of making the remaining display software a fixed reuseable unit, a sort of macro subroutine. This user interface is shown in Appendix B, exactly as presented to the user.

Finally, the software is sufficiently general enough to serve as a framework for use with other flow field measurement devices such as hot-wire anemometers or the four- and five-hole probes. Very little modification would be required to the existing software for these devices. In addition, a standard data file format, as shown in Appendix C, has been defined to standardize the acquisition and recording of experimental data for use in these programs.

III. Conclusions

In conclusion, the highly successful seven-hole probe flow survey technique has been improved through software modifications to allow:

a. interactive data collection in regions of interest

b. user-directed specified non-uniformly spaced data grids

c. the time required to collect and analyze data to be reduced

In essense, the process has become a computer-aided flow visualization system, user-oriented enough to allow one to perform wake surveys without expert intervention.

References

1. Gallington, R.W. "Measurement of Very Large Flow Angles with Non-Nulling Seven-Hole Probes." Aeronautics Digest - Spring/Summer 1980. USAFA-TR-80-17, USAF Academy, Colorado 80840.

2. Sisson, G. and Crandall, R. ""Canard Wake Measurement and Description." Aeronautics Digest - Fall/Winter 1980, USAFA-TR-81-4, USAF Academy, Colorado 80840.

3. Gerner, A.A. and Mauer, C.L. "Calibration of Seven-Hole Probes Suitable for High Angles in Subsonic Compressible Flows." Aeronautics Digest - Fall/Winter 1980, USAFA-TR-81-4, USAF Academy, Colorado 80840.

APPENDIX A

This appendix contains complete and documented listings for the following programs used in the seven-hole probe data acquistion system:

a. CMPSHP

b. MENU

c. GSUBS

d. COMCAL

e. FILEIO

f. GRAFS

```
01234567890123456789    ** RSX-11M V3.2 **    18-SEP-81    13:05:55    DF0:[152,13]CMPSHP.FTN;141    01234567890123456789
01234567890123456789    ** RSX-11M V3.2 **    18-SEP-81    13:05:55    DF0:[152,13]CMPSHP.FTN;141    01234567890123456789
01234567890123456789    ** RSX-11M V3.2 **    18-SEP-81    13:05:55    DF0:[152,13]CMPSHP.FTN;141    01234567890123456789
```

```
  CCCCCCCC  MM      MM  PPPPPPPP      SSSSSSSS  HH      HH  PPPPPPPP
  CCCCCCCC  MM      MM  PPPPPPPP      SSSSSSSS  HH      HH  PPPPPPPP
CC          MMMM  MMMM  PP      PP  SS          HH      HH  PP      PP
CC          MMMM  MMMM  PP      PP  SS          HH      HH  PP      PP
CC          MM  MM  MM  PP      PP  SS          HH      HH  PP      PP
CC          MM  MM  MM  PP      PP  SS          HH      HH  PP      PP
CC          MM      MM  PPPPPPPP      SSSSSS    HHHHHHHHHH  PPPPPPPP
CC          MM      MM  PPPPPPPP      SSSSSS    HHHHHHHHHH  PPPPPPPP
CC          MM      MM  PP                  SS  HH      HH  PP
CC          MM      MM  PP                  SS  HH      HH  PP
CC          MM      MM  PP                  SS  HH      HH  PP
CC          MM      MM  PP                  SS  HH      HH  PP            ....
  CCCCCCCC  MM      MM  PP          SSSSSSSS    HH      HH  PP            ....
  CCCCCCCC  MM      MM  PP          SSSSSSSS    HH      HH  PP            ....
                                                                          ....
```

```
FFFFFFFFFF  TTTTTTTTTT  NN      NN    ;;;;        11    44      44      11
FFFFFFFFFF  TTTTTTTTTT  NN      NN    ;;;;        11    44      44      11
FF              TT      NN      NN    ;;;;      1111    44      44    1111
FF              TT      NN      NN    ;;;;      1111    44      44    1111
FF              TT      NNNN    NN                11    44      44      11
FF              TT      NNNN    NN                11    44      44      11
FFFFFFFF        TT      NN  NN  NN    ;;;;        11    4444444444      11
FFFFFFFF        TT      NN  NN  NN    ;;;;        11    4444444444      11
FF              TT      NN    NNNN    ;;;;        11            44      11
FF              TT      NN    NNNN    ;;;;        11            44      11
FF              TT      NN      NN      ;;        11            44      11
FF              TT      NN      NN      ;;        11            44      11
FF              TT      NN      NN    ;;        111111          44    111111
FF              TT      NN      NN    ;;        111111          44    111111
```

```
01234567890123456789    ** RSX-11M V3.2 **    18-SEP-81    13:05:55    DF0:[152,13]CMPSHP.FTN;141    01234567890123456789
01234567890123456789    ** RSX-11M V3.2 **    18-SEP-81    13:05:55    DF0:[152,13]CMPSHP.FTN;141    01234567890123456789
01234567890123456789    ** RSX-11M V3.2 **    18-SEP-81    13:05:55    DF0:[152,13]CMPSHP.FTN;141    01234567890123456789
```

```
C*****************************************************************
C
C
C       COMPRESSIBLE FLOW SEVEN HOLE PROBE DATA ACQUISITION PROGRAM
C
C
C*****************************************************************
C
C       THIS PROGRAM IS DESIGNED TO PERFORM DATA ACQUISITION AND
C       REDUCTION OF DATA TO BE TAKEN WITH A SEVEN HOLE PROBE.
C
C       THE INPUTS AND OUTPUTS OF THE PROGRAM ARE AS FOLLOWS:
C
C       FILE 'DATAIN.DAT' IS THE INPUT DATA FILE.  IT CONTAINS ANY
C       DATA THAT MAY HAVE ALREADY BEEN TAKEN IN THE AXIAL LOCATION
C       PLANE TO WHICH WE WISH TO ADD MORE DATA.  THE DATA IS STORED IN
C       THE STANDARD DFAN DATA FILE FORMAT.  EACH RECORD HAS 20
C       DIMENSIONS AS DEFINED BELOW.  THERE ARE 100 DATA POINTS TO
C       EACH PLANE OF DATA CONSISTING OF 10 LINES OF 10 POINTS EACH.
C
C       FILE 'DATAOUT.DAT' IS THE OUTPUT DATA FILE.  IT WILL CONTAIN
C       EVERYTHING THAT WAS IN 'DATAIN.DAT' AND THE DATA THAT WILL
C       BE TAKEN DURING THIS RUN.  HOWEVER, ALL DATA POINTS THAT WERE
C       IN 'DATAIN.DAT' THAT WILL BE OBSCURED (COVERED UP/OVERLAYED)
C       BY LOCATIONS IN THE DATA TAKEN DURING THIS RUN WILL NOW BE
C       MARKED AS INVISIBLE.  THE DATA IS STORED IN THE SAME FORMAT AS
C       FOR 'DATAIN.DAT'.
C
C       FILE 'TARE.DAT' IS BOTH AN INPUT AND AN OUTPUT FILE, DEPENDING
C       ON THE RESPONSE OF THE USER.  IT WILL INITIALLY BE A NULL FILE
C       INDICATING THAT TARE DATA (AMPLIFIER ZERO CONDITIONS) MUST BE
C       TAKEN.  THIS DATA WILL BE OUTPUT TO A NON-NULL 'TARE.DAT' FILE
C       WHICH WILL BE USED THE NEXT TIME AROUND.  OLD TARE DATA MAY
C       BE IGNORED AND NEW TARE DATA TAKEN AT THE REQUEST OF THE USER.
C       FILE 'TARE.DAT' HAS ONLY ONE RECORD WITH 12 ENTRIES, ONE FOR
C       EACH INPUT/OUTPUT DATA VALUE.
C
C       FILE 'CPROBE.DAT' IS THE DATA FILE THAT CONTAINS THE CALIBRATION
C       COEFFICIENTS FOR THE COMPRESSIBLE SEVEN HOLE PROBE.  IS IS AN
C       INPUT FILE FOR THIS PROGRAM.
C
C       EACH DATA POINT IN 'DATAIN.DAT' AND 'DATAOUT.DAT' HAVE THE
C       FOLLOWING 20 DIMENSIONS IN THE ORDER SHOWN:
C
C       ENTRY   DESCRIPTION
C
C       1       Y LOCATION - INCHES
C       2       Z LOCATION - INCHES
C       3       VISIBLE (0=YES, 1=NO)
C       4       ALPHA - DEGREES
C       5       BETA - DEGREES
C       6       ALPHAT - DEGREES
C       7       BETAT - DEGREES
C       8       THETA - DEGREES
C       9       PHI - DEGREES
C       10      CTOTAL
C       11      CSTATIC
C       12      CA
C       13      CB
C       14      CQ
```

```
C        15       CZERO
C        16       CDYN
C        17       CM
C        18       RMACH
C        19       U VELOCITY COMPONENT
C        20       V VELOCITY COMPONENT
C
C
C***********************************************************************
C
C        DATA DECLARATIONS SECTION
C
C***********************************************************************
         DIMENSION DAT(20,100),IDLIST(20),XX(4),TDAT(20),YY(4),
     1     RSCR(50),ZERO(12),CAL(12),STRAP(12),
     2     KALPHA(7,20),KBETA(7,20),KQ(7,20),KZERO(7,20),KM(7,20)
         REAL KALPHA,KBETA,KQ,KZERO,KM
         LOGICAL*1 STITLE(20),LSCR(10,20),DLABEL(10,20),LTITLE(60)
         LOGICAL EOF
C*************************************************************************
C
C        DATA PRESETS AREA
C
C*************************************************************************
C        NAMES/LABELS FOR EACH DIMENSION
         DATA DLABEL/'Y',' ',' ',' ',' ',' ',' ',' ',' ',' ',
     1               'Z',' ',' ',' ',' ',' ',' ',' ',' ',' ',
     1               'V','I','S','I','B','L','E',' ',' ',' ',
     1               'A','L','P','H','A',' ',' ',' ',' ',' ',
     1               'B','E','T','A',' ',' ',' ',' ',' ',' ',
     1               'A','L','P','H','A','T',' ',' ',' ',' ',
     1               'B','E','T','A','T',' ',' ',' ',' ',' ',
     1               'T','H','E','T','A',' ',' ',' ',' ',' ',
     1               'P','H','I',' ',' ',' ',' ',' ',' ',' ',
     1               'C','T','O','T','A','L',' ',' ',' ',' ',
     1               'C','S','T','A','T','I','C',' ',' ',' ',
     1               'C','A',' ',' ',' ',' ',' ',' ',' ',' ',
     1               'C','B',' ',' ',' ',' ',' ',' ',' ',' ',
     1               'C','Q',' ',' ',' ',' ',' ',' ',' ',' ',
     1               'C','Z','E','R','O',' ',' ',' ',' ',' ',
     1               'C','D','Y','N',' ',' ',' ',' ',' ',' ',
     1               'C','M',' ',' ',' ',' ',' ',' ',' ',' ',
     1               'M','A','C','H',' ',' ',' ',' ',' ',' ',
     1               'U','-','V','E','L','O','C','I','T','Y',
     1               'V','-','V','E','L','O','C','I','T','Y'/
C        LIST OF ALL VARIABLES TO EXTRACT FROM INPUT DATA FILE
C        GET THEM ALL
         DATA IDLIST/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20/
         DATA YES/'Y'/
C        CALIBRATION COEFFICIENTS FOR THE AMPLIFIERS AND A/D CONVERTORS
C        THE ORDER IS AS FOLLOWS:  HORIZONTAL (Z) LOCATION, P1, P2, P3,
C        P4, P5, P6, P TOTAL, P STATIC, P7, VERTICAL (Y) LOCATION, TOTAL
C        TEMP, DESIRED VERTICAL LOCATION, DESIRED HORIZONTAL LOCATION
C
         DATA CAL/10.,1.2872,1.2229,1.2186,1.2007,1.2756,1.2367,
     1    1.1487,1.0955,1.2218,10.,1./
C        STRAPPING VOLTAGE FOR ALL AMPLIFIERS - ALL AT +/- 1 VOLT
         DATA STRAP/12*1./
C        NUMBER OF DIMENSIONS IN A DATA POINT IS 20
         DATA ND/20/
         DATA EOF/.FALSE./
```

```
C######################################################################
C
C        FORMAT STATEMENTS
C
C#################v####################################################
C        FORMAT STATEMENT FOR FIRST HEADER RECORD OF STD FILE FORMAT
 11      FORMAT(5I5)
 12      FORMAT(' DO YOU WANT TO RETAKE TARE/AMPLIFIER ZERO DATA? [Y/N]')
 13      FORMAT(A1)
 14      FORMAT(' THE PRESENT DATA TITLE IS:'/2X,40A1/
     1    ' DO YOU WISH TO CHANGE IT?')
 15      FORMAT(' ENTER DATA TITLE (40 CHARS MAX)')
 16      FORMAT(40A1)
 17      FORMAT(' START TUNNEL AND ENTER RETURN WHEN READY')
 18      FORMAT(' ENTER ATMOSPHERIC PRESSURE (IN-HG)')
 19      FORMAT(F10.0)
 20      FORMAT(' COORDINATE PAIRS ARE (VERT,HORIZ)')
 21      FORMAT(2F10.0)
 22      FORMAT(' ENTER LOCATION OF LOWER LEFT AND ENTER RETURN')
 23      FORMAT(' ENTER LOCATION OF UPPER LEFT AND ENTER RETURN')
 24      FORMAT(' ENTER LOCATION OF LOWER RIGHT AND ENTER RETURN')
 25      FORMAT(' ENTER LOCATION OF UPPER RIGHT AND ENTER RETURN')
 26      FORMAT(' LOWER LEFT:',2F10.3,/,' UPPER LEFT:',2F10.3,/,
     1           ' LOWER RIGHT:',2F10.3,/,' UPPER RIGHT:',2F10.3,/,
     2           ' ARE THESE LOCATIONS CORRECT?')
C######################################################################
C
C        MAIN PROGRAM CODE STARTS HERE
C
C######################################################################
C        GET THE CALIBRATION COEFFICIENTS FILE ASSIGNED TO THE PROGRAM
         CALL ASSIGN(2,'[152,2]CPROBE.DAT',17)
         CALL FDBSET(2,'READONLY',,,7)
         DEFINE FILE 2(7,256,U,IREC)
C        GET ALL OF THE CALIBRATION COEFFICIENTS
         IREC=1
         DO 1000 I=1,7
         READ(2'IREC)(KALPHA(I,J),J=1,20),
     1               (KBETA(I,J),J=1,20),
     2               (KQ(I,J),J=1,20),
     3               (KZERO(I,J),J=1,20),
     4           (KM(I,J).J=1,20)
 1000    CONTINUE
C        WE ARE DONE WITH THE CALIBRATION COEFFICIENTS DATA FILE
C        CLOSE AND RELEASE IT
         CALL CLOSE(2)
C        GET THE LPS-11 LABORATORY PERIPHERIAL SYSTEM ASSIGNED TO UNIT 1
         CALL ASLSLN(1)
C        GET THE INPUT DATA FILE ASSIGNED AND ALLOCATED TO THE PROGRAM
         CALL ASSIGN(2,'DATAIN.DAT',10)
         CALL FDBSET(2,'OLD','SHARE')
C        GET THE OUTPUT DATA FILE ASSIGNED AND ALLOCATED TO THE PROGRAM
         CALL ASSIGN(3,'DATAOUT.DAT',11)
         CALL FDBSET(3,'NEW','SHARE')
C        GET THE TARE/AMPLIFIER ZERO DATA FILE ASSIGNED AND ALLOCATED
         CALL ASSIGN(4,'TARE.DAT',8)
         CALL FDBSET(4,'OLD','SHARE')
C        READ THE TARE DATA FROM THE FILE - IF END OF FILE IS ENCOUNTERED
C        THEN WE MUST TAKE TARE DATA - IF NOT THEN TARE DATA IS OPTIONAL
C        AT THE REQUEST OF THE USER
         READ(4,END=1100)ZERO
```

```
C       GIVE THE USER THE OPTION TO RETAKE THE TARE DATA
        WRITE(5,12)
        READ(5,13)ATARE
C       IF THE ANSWER WAS NOT YES THEN SKIP TAKING TARE DATA AGAIN
        IF (ATARE.NE.YES) GO TO 1200
C       TAKE TARE/AMPLIFIER ZERO DATA NOW
 1100   CONTINUE
        CALL TARE(ZERO,STRAP)
C       WE NO LONGER WANT THE INPUT TARE FILE SO CLOSE AND RELEASE IT
C       SINCE IT IS NOW OBSOLETE WE WILL WANT TO MAKE A NEW ONE
        CALL CLOSE(4)
C       ASSIGN AND CREATE A NEW TARE DATA FILE TO CONTAIN THE JUST
C       ACQUIRED AMPLIFIER ZERO DATA
        CALL ASSIGN(4,'TARE.DAT',8)
        CALL FDBSET(4,'NEW','SHARE')
C       WRITE THE NEW DATA TO DISK
        WRITE(4)ZERO
 1200   CONTINUE
C       NOW WE ARE THRU WITH THE TARE FILE COMPLETELY SO RELEASE IT
        CALL CLOSE(4)
C       IT IS NOW TIME TO SEE IF THE INPUT DATA FILE WAS A NULL FILE
C       OR IF THIS RUN IS ADDING DATA TO AN EXISTING FILE.  IF IT
C       IS A NEW RUN THEN THE TITLE MUST BE INPUT, IF IT IS A
C       CONTINUATION RUN THEN WE WILL TELL THE USER THE TITLE AND
C       ALLOW HIM TO CHANGE IT IF HE SO DESIRES.
C
C       READ THE FIRST RECORD - IF END OF FILE THEN GO TO GET TITLE
        READ(2,11,END=1300)MND,NP,NL,NG,NLAT
C       GET THE REST OF THE HEADER RECORDS FROM THIS GRID, INCLUDING
C       THE TITLES.
        CALL GETHDR(2,MND,ND,IDLIST,LSCR,DLABEL,STITLE,LTITLE,EOF)
C       OUTPUT THE TITLE TO THE USER - SEE IF HE WANTS TO CHANGE IT
        WRITE(5,14)(LTITLE(J),J=1,40)
        READ(5,13)ATITL
C       IF THE USER DOESN'T WANT TO CHANGE IT THEN GO TO THEN NEXT STEP
        IF (ATITL.NE.YES) GO TO 2000
 1300   CONTINUE
C       ASK FOR AND READ IN THE 40 CHARACTERS FOR THE TITLE
        WRITE(5,15)
        READ(5,16)(LTITLE(I),I=1,40)
C       MAKE THE TIME AND DATA THE LAST 20 CHARACTERS OF THE 60
C       CHARACTER TITLE ARRAY
        CALL TIME(LTITLE(41))
        CALL DATE(LTITLE(49))
 2000   CONTINUE
C       TELL THE USER TO START THE TUNNEL AND HIT 'RETURN' KEY
C       WHEN HE IS READY TO CONTINUE
        WRITE(5,17)
C       CLOSE THE INPUT DATA FILE AND GET IT REASSIGNED SO WE ARE
C       BACK TO THE START OF THE FILE.  NORMALLY THIS WOULD SIMPLY
C       BE A REWIND STATEMENT BUT THAT ISN'T WORKING ON SEQUENTIAL
C       DISK FILES ON THIS SYSTEM AT THE PRESENT TIME.  THIS DOES
C       THE SAME THING.
        CALL CLOSE(2)
        CALL ASSIGN(2,'DATAIN.DAT',10)
        CALL FDBSET(2,'OLD','SHARE')
C       NOW WE WAIT FOR THE USER TO SIGNAL THAT HE IS READY TO CONTINUE
        READ(5,13)AGO
C       NOW WE ASK FOR THE ATMOSPHERIC PRESSURE READING IN INCHES OF
C       MERCURY WHICH WE WILL THEN CONVERT TO PSIA USING THE CONSTANT
C       FOR MERCURY AT THIS ELEVATION AND TEMPERATURE.
```

```
          WRITE(5,13)
          READ(5,19)PATM
          PATM=PATM*.4892
 2100     CONTINUE
C         TELL THE USER THAT COORDINATES ARE TO BE ENTERED AS PAIRS
C         IN THE ORDER (VERTICAL,HORIZONTAL) WHERE THE COORDINATE IS
C         A LOCATION IN INCHES FROM THE USER AND TRAVERSE DESIGNATED
C         ORIGIN.
          WRITE(5,20)
C         ASK FOR COORDINATES FOR THE LOWER LEFT CORNER OF THE PLANE
          WRITE(5,22)
          READ(5,21)XX(1),YY(1)
C         ASK FOR COORDINATES FOR THE UPPER LEFT CORNER OF THE PLANE
          WRITE(5,23)
          READ(5,21)XX(2),YY(2)
C         ASK FOR COORDINATES FOR THE LOWER RIGHT CORNER OF THE PLANE
          WRITE(5,24)
          READ(5,21)XX(3),YY(3)
C         ASK FOR COORDINATES FOR THE UPPER RIGHT CORNER OF THE PLANE
          WRITE(5,25)
          READ(5,21)XX(4),YY(4)
C         NOW THAT WE HAVE THE COORDINATES ASK THE USER TO VERIFY THAT HE
C         HAS ENTERED THEM CORRECTLY.
          WRITE(5,26)((XX(I),YY(I)),I=1,4)
          READ(5,13)AOK
C         IF THE COORDINATES ARE NOT RIGHT THEN GO BACK AND TRY AGAIN
          IF (AOK.NE.YES)GO TO 2100
 3000     CONTINUE
C         NOW WE PROCESS THE DATA THAT EXISTS (IF ANY) ON THE INPUT DATA
C         FILE.  EACH PLANE/GRID OF DATA IS PROCESSED AND ANY DATA POINTS
C         THAT ARE COVERED BY THE NEWLY DEFINED PLANE/GRID WILL BE MARKE
C         AS INVISIBLE.  THEN THE DATA WILL BE OUTPUT TO THE OUTPUT DATA
C         FILE TO WHICH THE NEW DATA WILL BE APPENDED A LITTLE LATER.
C
C         GET A GRID/PLANE OF DATA, 20 DIMENSIONS PER POINT, 100 POINTS
          CALL GETGRD(2,ND,IDLIST,100,LSCR,RSCR,LSCR,LSCR,DLABEL,
     1     DAT,NP,NL,NG,NLAT,EOF)
C         IF THERE IS NO MORE DATA TO PROCESS THEN CONTINUE TO THE NEXT
C         STEP IN THE PROCESS - GO TO ACQUIRE THE NEW PLANE OF DATA
          IF(EOF)GO TO 3100
C         DETERMINE IF ANY POINTS IN THIS PLANE SHOULD BE MARKED AS
C         INVISIBLE BECAUSE THEY ARE CONTAINED IN THE NEW PLANE.
C         THIS ROUTINE MARKS POINTS INVISIBLE AS REQUIRED.
          CALL CNTAIN(NP,NL,DAT,XX,YY)
C         NOW THAT ANY POINTS THAT ARE CONTAINED HAVE BEEN MARKED AS
C         INVISIBLE, WE OUTPUT THE PLANE OF DATA TO THE OUTPUT FILE
C         AND SEE IF THERE IS MORE DATA ON THE INPUT FILE TO PROCESS.
          CALL PUTLIN(3,ND,NP,NL,NG,NLAT,STITLE,LTITLE,DLABEL,DAT)
          GO TO 3000
 3100     CONTINUE
C         NOW WE HAVE PROCESSED ALL THE DATA FROM THE INPUT FILE AND ARE
C         ALMOST READY TO TAKE THE DATA FOR THIS NEW PLANE.  FIRST WE
C         DEFINE OUR PLANE AS 10 HORIZONTAL LINES OF 10 POINTS EACH.
          NP=10
          NL=10
C         NOW WE MARK ALL POINTS IN THE NEW PLANE AS VISIBLE.
          DO 3200 I=1,100
          DAT(3,I)=0.
 3200     CONTINUE
C         NOW WE USE ROUTINE BLINE TO DETERMINE THE LOCATION OF
C         EACH DATA POINT IN THE PLANE.
```

```
      CALL BLINE(NP,NL,DAT,XX,YY)
C     NOW WE TAKE THE DATA FOR THIS PLANE - THE DATA ACQUISITION
C     AND REDUCTION CALLS ARE PART OF THE TAKRED ROUTINE.
      CALL TAKRED(NP,NL,DAT,ZERO,STRAP,CAL,PATM,
     1  KALPHA,KBETA,KQ,KZERO,KM)
C     NOW THAT THE DATA FOR THIS PLANE HAS BEEN TAKEN AND REDUCED,
C     WE OUTPUT IT TO THE OUTPUT DATA FILE WITH THE OTHER DATA
C     ALREADY THERE FROM THE INPUT DATA FILE.
      CALL PUTLIN(3,ND,NP,NL,NG+1,NLAT,STITLE,LTITLE,DLABEL,DAT)
C     WE NO LONGER NEED THE INPUT DATA FILE SO RELEASE IT.
      CALL CLOSE(2)
C     WE NO LONGER NEED THE OUTPUT DATA FILE SO RELEASE IT TOO.
      CALL CLOSE(3)
C     WE ARE THRU WITH THE DATA ACQUISITION AND REDUCTION OF A PLANE
C     OF DATA SO TERMINATE THE PROGRAM.
      STOP
      END
C*************************************************************
C
C
C     THIS ROUTINE IS USED TO TAKE THE AMPLIFIER ZERO DATA
C     FOR ALL THE TRANSDUCERS.
C
C
C**************************************************************
      SUBROUTINE TARE(ZERO,STRAP)
C     ARRAY ZERO WILL CONTAIN THE AMPLIFIER ZERO VOLTAGES
C     ARRAY STRAP CONTAINS THE STRAPPING VOLTAGE FOR EACH CHANNEL
      DIMENSION ZERO(12),STRAP(12),DATA(12)
C     CLEAR THE AMPLIFIER ZERO ARRAY
      DO 10 I=1,12
      ZERO(I)=0.
 10   CONTINUE
C     CALL THE DATA ACQUISITION SUBROUTINE - IT ACQUIRES THE
C     ACTUAL VOLTAGES FOR EACH ANALOG TO DIGITAL CONVERSION
C     CHANNEL THAT WE ARE INTERESTED IN.
      CALL TAKDAT(DATA,ZERO,STRAP)
C     THERE ARE CERTAIN CHANNELS THAT DO NOT USE AMPLIFIERS
C     AND FOR WHICH THE AMPLIFIER ZERO ENTRY MUST BE ZERO.
C     THESE ARE AS FOLLOWS:
C     ACTUAL HORIZONTAL LOCATION
C     ACTUAL VERTICAL LOCATION
C     DESIRED HORIZONTAL LOCATION
C     DESIRED VERTICAL LOCATION
C     TEMPERATURE
C     MOVE THE AMPLIFIER ZERO VALUES THAT WE NEED INTO THE ARRAY
      DO 20 I=2,10
      ZERO(I)=DATA(I)
 20   CONTINUE
      RETURN
      END
C***************************************************************
C
C
C     THIS ROUTINE COMPUTES THE DESIRED LOCATION OF ALL 100 DATA
C     POINTS BASED ON THE LOCATION OF THE 4 CORNERS DEFINED IN
C     ARRAYS XX AND YY.  THE DATA POINTS ARE FROM MINIMUM TO
C     MAXIMUM HORIZONTAL VALUE FOR EACH LINE AND EACH LINE IS
C     FOR A GREATER VERTICAL VALUE THAN THE PREVIOUS LINE
C
C
```

```
C###################################################################
        SUBROUTINE BLINE(N,M,DAT,XX,YY)
C       N IS THE NUMBER OF POINTS PER LINE
C       M IS THE NUMBER OF LINES PER PLANE
C       DAT IS THE DATA ARRAY
C       XX IS THE 4 HORIZONTAL CORNER LOCATIONS ARRAY
C       YY IS THE 4 VERTICAL CORNER LOCATIONS ARRAY
        DIMENSION DAT(20,100),XX(4),YY(4)
C       INITIALIZE THE DATA POINT INDEX
        K=1
C       COMPUTE THE HORIZONTAL INCREMENT
        UINC=1./FLOAT(N-1)
C       COMPUTE THE VERTICAL INCREMENT
        VINC=1./FLOAT(M-1)
C       CLEAR THE VERTICAL INCREMENT VALUE
        V=0.
C       FOR EACH LINE IN THE PLANE PROCESS THE LOCATIONS
        DO 20 J=1,M
C       CLEAR THE HORIZONTAL INCREMENT VALUE
        U=0.
C       FOR EACH POINT ON A LINE - COMPUTE THE LOCATION
        DO 10 I=1,N
C       COMPUTE THE VERTICAL LOCATION
        DAT(1,K)=(1-U)*(1-V)*XX(1)+(1-U)*V*XX(2)
     1       +U*(1-V)*XX(3)+U*V*XX(4)
C       COMPUTE THE HORIZONTAL LOCATION
        DAT(2,K)=(1-U)*(1-V)*YY(1)+(1-U)*V*YY(2)
     1       +U*(1-V)*YY(3)+U*V*YY(4)
C       INCREMENT THE DATA POINT INDEX
        K=K+1
C       INCREMENT THE HORIZONTAL INCREMENT VALUE
        U=U+UINC
 10     CONTINUE
C       INCREMENT THE VERTICAL INCREMENT VALUE
        V=V+VINC
 20     CONTINUE
        RETURN
        END
C###############################################################
C
C
C       THIS ROUTINE DETERMINES IF EACH OF THE 100 DATA POINTS
C       IS CONTAINED IN THE BOUNDARYS OF THE NEWLY DEFINED PLANE.
C
C
C###############################################################
        SUBROUTINE CNTAIN(N,m,DAT,XX,YY)
C       N IS THE NUMBER OF POINTS PER LINE
C       M IS THE NUMBER OF LINES PER PLANE
C       DAT IS THE DATA ARRAY
C       XX IS THE HORIZONTAL CORNER LOCATIONS ARRAY
C       YY IS THE VERTICAL CORNER LOCATION ARRAY
        DIMENSION DAT(20,100),XX(4),YY(4)
        LOGICAL NAREA
C       SET UP THE DATA POINT INDEX
        K=1
C       FOR EACH LINE IN THE PLANE PROCESS THE DATA POINTS
        DO 10 J=1,M
C       FOR EACH POINT IN A LINE PROCESS THE DATA POINT
        DO 10 I=1,N
C       COMPUTE THE SLOPE AND INTERCEPT INFORMATION FOR
```

```
C       THE LINES THAT MAKE UP THE FOUR SIDES OF THE PLANE.
        A1=(XX(1)-XX(3))/(YY(1)-YY(3))
        A2=(XX(2)-XX(4))/(YY(2)-YY(4))
        A3=(YY(1)-YY(2))/(XX(1)-XX(2))
        A4=(YY(3)-YY(4))/(XX(3)-XX(4))
        B1=XX(1)-A1*YY(1)
        B2=XX(2)-A2*YY(2)
        B3=YY(1)-A3*XX(1)
        B4=YY(3)-A4*XX(3)
C       SEE IF THE DATA POINT IS TO THE LEFT OF THE PLANE
        X=A1*DAT(2,K)+B1
C       IF SO THEN IT IS NOT CONTAINED
        IF (X.GE.DAT(1,K))GO TO 10
C       SEE IF THE DATA POINT IS TO THE RIGHT OF THE PLANE
        X=A2*DAT(2,K)+B2
C       IF SO THEN IT IS NOT CONTAINED
        IF (X.LE.DAT(1,K))GO TO 10
C       SEE IF THE DATA POINT IS BELOW THE PLANE
        X=A3*DAT(1,K)+B3
C       IF SO THEN IT IS NOT CONTAINED
        IF (X.GE.DAT(2,K))GO TO 10
C       SEE IF THE DATA POINT IS ABOVE THE LINE
        X=A4*DAT(1,K)+B4
C       IF SO THEN IT IS NOT CONTAINED
        IF (X.LE.DAT(2,K))GO TO 10
C       THE DATA POINT MUST BE INSIDE THE PLANE - MARK AS INVISIBLE
        DAT(3,K)=1.
 10     K=K+1
        RETURN
        END
C**********************************************************************
C
C
C       THIS ROUTINE CONTROLS THE ACTUAL DATA ACQUISITION AND REDUCTION
C       OF DATA POINTS TAKEN WITH THE SEVEN HOLE PROBE IN COMPRESSIBLE
C       FLOW.
C
C
C***********************************************************************
        SUBROUTINE TAKRED(N,M,DAT,ZERO,STRAP,CAL,PATM,
     1    KALPHA,KBETA,KQ,KZERO,KM)
C       N IS THE NUMBER OF POINTS PER LINE
C       M IS THE NUMBER OF LINES PER PLANE
C       DAT IS THE DATA STORAGE ARRAY
C       ZERO IS THE TARE/AMPLIFIER ZERO DATA ARRAY
C       STRAP IS THE AMPLIFIER STRAPPING VOLTAGE ARRAY
C       CAL IS THE CALIBRATION CONSTANTS ARRAY
C       KALPHA, KBETA, KQ, KZERO, KM ARE THE SEVEN HOLE
C       PROBE CALIBRATION COEFFICIENTS FOR COMPRESSIBLE FLOW.
C
        DIMENSION DATA(12),ISB(2)
        DIMENSION DAT(20,100),ZERO(12),STRAP(12),CAL(12),
     1    KM(7,20),KALPHA(7,20),KBETA(7,20),KZERO(7,20),KQ(7,20)
        REAL KALPHA,KBETA,KZERO,KQ,KM
C       THE ISERP FUNCTION COMPUTES THE INDEX OF A DATA POINT SO THAT
C       THE TRAVERSE TRAVEL FROM ONE POINT TO THE NEXT IS MIMINIZED.
        ISERP(M,N)=MOD(M+1,2)*(10*M+1-N)+MOD(M,2)*(10*M-10+N)
C       OUTPUT THE LOCATION OF THE FIRST DATA POINT FOR THE USER
C       THIS WAY HE CAN CHECK THAT ALL IS WELL BEFORE TURNING
C       THE TRAVERSE MECHANISM FROM MANUAL TO AUTO
        WRITE(5,807)DAT(1,1),DAT(2,1)
 807     FORMAT(' FIRST DATA POINT: VERTICAL=',F10.3,
     1          ' HORIZONTAL=',F10.3)
C        FOR EACH LINE OF DATA IN THE PLANE - PROCESS THE POINTS
         DO 20 J=1,M
C        FOR EACH POINT IN A LINE - PROCESS THE POINT
         DO 20 I=1,N
C        GET INDEX OF THE NEXT DATA POINT
         K=ISERP(J,I)
C        MOVE THE TRAVERSE MECHANISM TO THE DESIRED LOCATION
         CALL MOVE(DAT(2,K),DAT(1,K))
C        WAIT 1 SECOND FOR CONDITIONS TO STABILIZE
         CALL WAIT(1,2,ISB)
C        CALL THE DATA ACQUISITION SUBROUTINE TO ACTUALLY READ THE DATA
         CALL TAKDAT(DATA,ZERO,STRAP)
         DO 10 II=1,12
 10      DATA(II)=DATA(II)*CAL(II)
C        REPLACE THE DESIRED LOCATION WITH THE ACTUAL LOCATION
         DAT(1,K)=DATA(1)
         DAT(2,K)=DATA(11)
C        REDUCE THE DATA TAKEN WITH THE COMPRESSIBLE SEVEN HOLE PROBE
         CALL COMCAL(0,DATA,KALPHA,KBETA,KQ,KZERO,KM,PATM,DAT(4,K),
     1    DAT(5,K),DAT(6,K),DAT(7,K),DAT(8,K),DAT(9,K),DAT(10,K),
     2    DAT(11,K),DAT(12,K),DAT(13,K),DAT(14,K),DAT(15,K),DAT(16,K),
     3    DAT(17,K),DAT(18,K),DAT(19,K),DAT(20,K))
 20      CONTINUE
         RETURN
         END
C*********************************************************************
C
C
C        THIS ROUTINE DOES THE ACTUAL DATA ACQUISITION THRU THE LPS-11
C        ANALOG TO DIGITAL CONVERTORS.
C
C
C*********************************************************************
         SUBROUTINE TAKDAT(X,ZCF,STRAP)
         DIMENSION X(12),STRAP(12),ZCF(12)
C        FOR EACH OF THE 12 CHANNELS OF DATA TO BE ACQUIRED
       DO 2000 I1=1,12
       I=I1
C        IF THIS IS THE ELEVENTH CHANNEL THEN ACTUALLY USE CHANNEL 0
       IF (I1.EQ.11)I=0
C        SET THE DATA TO ZERO INITIALLY
         RDATA=0.
C        TAKE 10 SAMPLES AND AVERAGE THEN
         DO 1900 J=1,10
C        GET THE VALUE OF THE VOLTAGE IN DIGITAL FORM
 1500    CONTINUE
         CALL ADC(I,DATA)
C        CONVERT IT TO A VOLTAGE
         DATOUT=((((DATA/64.)/2047.5)*STRAP(I1))-STRAP(I1))
C        CHECK FOR DATA OUT OF RANGE
         DATA=DATA/64.
         IF ((DATA.GE.4095.).OR.(DATA.LE.0.))GO TO 1700
C        ADVISE THAT THE CHANNEL VOLTAGE VALUE WAS OUT OF RANGE
         WRITE(5,1600) I
         GO TO 1500
 1600 FORMAT(1X,'TAKDAT:  CHANNEL ',O2,' OUT OF RANGE')
 1700 CONTINUE
         RDATA=RDATA+DATOUT
 1900    CONTINUE
```

```
      DATOUT=RDATA/10.
      X(I1)=DATOUT-ZCF(I1)
 2000 CONTINUE
      RETURN
      END
C*******************************************************************
C     THIS SUBROUTINE EXPECTS THE HORIZONTAL TRAVERSE POSITION
C     ON A/D INPUT CHANNEL 0 AND THE VERTICAL TRAVERSE POSITION
C     ON A/D INPUT CHANNEL 1.  THE 'H' OUTPUT CABLE CONNECTS TO
C     THE HORIZONTAL DRIVE AND THE 'V' OUTPUT CABLE CONNECTS TO
C     THE VERTICAL DRIVE.  THIS CHANGE AND COMMENTS ADDED BY
C     CAPTS BOLICK AND SISSON ON 27 MAY 80.
C*******************************************************************
      SUBROUTINE MOVE(Z,Y)
      DIMENSION V(2),P(3),DEL(2),IBUF(8),IRATE(2)
      DIMENSION ISB(2),FL(2),CA(2),CB(2),VC(2),AV(2)
C     SET THE LIMIT FOR USING MAXIMUM DRIVING VOLTAGE TO 3 INCHES
C     FOR BOTH HORIZONTAL AND VERTICAL
      CA(1)=3.
      CA(2)=3.
C     SET FRACTIONS FOR COMPUTING DRIVING VOLTAGES INSIDE OF 3 INCHES
      CB(1)=9./15.
      CB(2)=9./15.
      VC(1)=13./15.
      VC(2)=13./15.
C     SET DESIRED HORIZONTAL POSITION
      P(1)=Z
C     SET DESIRED VERTICAL POSITION
      P(2)=Y
C     SET THE COMPLETE FLAG FOR BOTH TO ZERO
      DO 5 I=1,2
      FL(I)=0.
 5    CONTINUE
C     CHECK LOCATION AND SET DRIVE VOLTAGE FOR BOTH
  15  DO 35 I=1,2
C     THE ANALOG CHANNELS ARE 0 AND 1 FOR HORIZONTAL AND VERTICAL
C     AXIS DRIVES RESPECTIVELY
     I1=I-1
C     SET CURRENT POSITION TO ZERO
      V(I)=0.
C     TAKE A 4 TIME AVERAGE SAMPLE OF THE LOCATION
      DO 16 IA=1,4
      CALL ADC(I1,AV(I),1,ISB)
   16 V(I)=V(I)+AV(I)/131072.
      V(I)=V(I)/4.
C     IF AN ERROR WAS ENCOUNTERED ADVISE THE USER
      IF(ISB(1).NE.1)WRITE(5,908)ISB(1)
C     COMPUTE THE DELTA DISTANCE BETWEEN ACTUAL AND DESIRED LOCATIONS
      DEL(I)=-((V(I)/1.-1.)*10.-P(I))
C     IF GREATER THAN +3 INCHES AWAY - USE MAX POSITIVE VOLTAGE
      IF(DEL(I).GT.CA(I))GO TO 20
C     IF BETWEEN 0 AND + 3 USE PROPORTIONAL VOLTAGE BASED ON DELTA
      IF(DEL(I).GT.0.)GO TO 25
C     IF BETWEEN 0 AND - 3 USE PROPORTIONAL VOLTAGE BASED ON DELTA
      IF(DEL(I).GT.-CA(I))GO TO 30
C     MUST BE GREATER THAN - 3 INCHES AWAY, SO USE MAX NEGATIVE DRIVE
      IBUF(I+4)=INT(2047.5-VC(I)*2047.5)
      GO TO 35
  20  IBUF(I+4)=INT(2047.5+VC(I)*2047.5)
      GO TO 35
  25  IBUF(I+4)=INT((1.+CB(I)+(VC(I)-CB(I))*DEL(I)/CA(I))*2047.5)
```

```
        GO TO 33
  30    IBUF(I+4)=INT((1.-CB(I)+(VC(I)-CB(I))*DEL(I)/CA(I))*2047.5)
C       IF THE DELTA DISTANCE IS LESS THAN .01 INCHES THEN WE
C       MUST SET THE COMPLETE FLAG AND STOP THIS DRIVE
  33    IF(ABS(DEL(I)).GT..01)GO TO 35
        FL(I)=1.
        IBUF(I+4)=2048
  35    CONTINUE
C       SET UP TO DRIVE THE DIGITAL TO ANALOG CONVERTORS ON THE LPS-11
        IRATE(1)=2
        IRATE(2)=1
  36    IBUF(7)=IBUF(5)
        IBUF(8)=IBUF(6)
C       OUTPUT THE VALUE TO THE D-TO-A DRIVERS
        CALL SDAC(IBUF,8,16,IRATE,7,0,2,ISB,1)
C       IF AN ERROR WAS ENCOUNTERED - TELL THE USER
        IF(ISB(1).GT.1)WRITE(5,907)ISB(1)
C       IF BOTH COMPLETE FLAGS ARE SET THEN CHECK FOR ACTUAL COMPLETION
        IF(FL(1)+FL(2).EQ.2) GO TO 55
C       GO AND SEE IF WE ARE CLOSE ENOUGH TO STOP YET
        GO TO 15
C       CHECK ON THE FINAL POSITION AGAIN AND CLEAR COMPLETE FLAGS
  55    DO 75 I=1,2
      I1=I-1
        CALL ADC(I1,V(I),1,ISB)
        DEL(I)=-((V(I)/131072.-1.)*10.-P(I))
  75    FL(I)=0.
C       IF BOTH ARE WITHIN .01 THEN THEY ARE WITHIN LIMITS
        IF(ABS(DEL(1)).LT..01.AND.ABS(DEL(2)).LT..01)GO TO 56
C       AT LEAST ONE IS OUT OF LIMITS - GO MOVE IT AGAIN
        GO TO 15
   56 CONTINUE
 907    FORMAT(' SDAC ERROR ',I4)
 908    FORMAT(' MOVE ADC ERROR ',I4)
C       INSURE THAT BOTH DRIVES ARE STOPPED
      IF (IBUF(5).NE.2048)GO TO 910
      IF (IBUF(6).EQ.2048)GO TO 999
  910 CONTINUE
C       AT LEAST ONE DRIVE WAS NOT STOPPED - STOP THEM AND
C       GO TO RECHECK THE POSITIONS
      IBUF(5)=2048
      IBUF(6)=2048
      IBUF(7)=2048
      IBUF(8)=2048
      CALL SDAC(IBUF,8,16,IRATE,7,0,2,ISB,1)
      GO TO 15
  999 CONTINUE
C       BOTH DRIVES WERE STOPPED BUT WE WILL INSURE IT BY SETTING
C       THE DRIVE VOLTAGES TO ZERO ONE LAST TIME
      CALL SDAC(IBUF,8,16,IRATE,7,0,2,ISB,1)
        RETURN
        END
```

```
01234567890123456789     ** RSX-11M V3.2 **     18-SEP-81     14:21:49     DP0:[152,13]MENU.FTN;42     01234567890123456789
01234567890123456789     ** RSX-11M V3.2 **     18-SEP-81     14:21:49     DP0:[152,13]MENU.FTN;42     01234567890123456789
01234567890123456789     ** RSX-11M V3.2 **     18-SEP-81     14:21:49     DP0:[152,13]MENU.FTN;42     01234567890123456789



MM      MM   EEEEEEEEEE   NN      NN   UU      UU
MM      MM   EEEEEEEEEE   NN      NN   UU      UU
MMMM  MMMM   EE           NN      NN   UU      UU
MMMM  MMMM   EE           NN      NN   UU      UU
MM  MM  MM   EE           NNNN    NN   UU      UU
MM  MM  MM   EE           NNNN    NN   UU      UU
MM      MM   EEEEEEEE     NN  NN  NN   UU      UU
MM      MM   EEEEEEEE     NN  NN  NN   UU      UU
MM      MM   EE           NN    NNNN   UU      UU
MM      MM   EE           NN    NNNN   UU      UU
MM      MM   EE           NN      NN   UU      UU      ....
MM      MM   EE           NN      NN   UU      UU      ....
MM      MM   EEEEEEEEEE   NN      NN   UUUUUUUUUU      ....
MM      MM   EEEEEEEEEE   NN      NN   UUUUUUUUUU      ....



FFFFFFFFFF   TTTTTTTTTT   NN      NN      ;;;;     44      44    222222
FFFFFFFFFF   TTTTTTTTTT   NN      NN      ;;;;     44      44    222222
FF               TT       NN      NN      ;;;;     44      44  22      22
FF               TT       NN      NN      ;;;;     44      44  22      22
FF               TT       NNNN    NN               44      44          22
FF               TT       NNNN    NN               44      44          22
FFFFFFFF         TT       NN  NN  NN      ;;;;     4444444444        22
FFFFFFFF         TT       NN  NN  NN      ;;;;     4444444444        22
FF               TT       NN    NNNN      ;;;;             44      22
FF               TT       NN    NNNN      ;;;;             44      22
FF               TT       NN      NN        ;;             44    22
FF               TT       NN      NN        ;;             44    22
FF               TT       NN      NN      ;;               44  2222222222
FF               TT       NN      NN      ;;               44  2222222222



01234567890123456789     ** RSX-11M V3.2 **     18-SEP 81     14:21:49     DP0:[152,13]MENU.FTN;42     01234567890123456789
01234567890123456789     ** RSX-11M V3.2 **     18-SEP-81     14:21:49     DP0:[152,13]MENU.FTN;42     01234567890123456789
01234567890123456789     ** RSX-11M V3.2 **     18-SEP-81     14:21:49     DP0:[152,13]MENU.FTN;42     01234567890123456789
```

```
C
C
C       THIS PROGRAM IS USED TO BUILD THE 'MENU.DAT'
C       DATA FILE REQUIRED BY 'GRAFS'.  IF A MENU FILE
C       ALREADY EXISTS THEN IT READS THAT AND WORKS
C       FROM THERE.  OTHERWISE IS ASSUMES CERTAIN DEFAULTS.
C       THE USER IS ALLOWED TO MODIFY ANY PARAMETER SETTINGS
C       BEFORE THE DATA IS REWRITTEN TO DISK.
C
C
C###########################################################
C
C
C       DATA DECLARATIONS AREA
C
C
C###############################################################
        LOGICAL*1 DLAB(10,5)
        LOGICAL*1 DLABEL(10,20),LSCR(200),STITLE(20),LTITLE(60),TITLE(60
        LOGICAL NOCHG,EOF,MNMX,DEFALT
        DIMENSION IDLIST(5),DAT(5,100),X(50)
C###############################################################
C
C
C       FORMAT DECLARATIONS AREA
C
C
C####################################################################
 101    FORMAT(' ENTER PLOT OPTION: 0=CONTOUR, 1=AXONOMETRIC,',
     1          ' 2=CROSS FLOW')
 102    FORMAT(F10.0)
 103    FORMAT(' YOU MAY PLOT ANY OF THE FOLLOWING QUANTITIES:')
 104    FORMAT(5(1X,I2,1X,10A1))
 105    FORMAT(' 1.OPTION = ',F2.0,/,
     1          ' 2.TITLE = ',60A1,/,
     1          ' 3.X AXIS ANGLE = ',F10.3,/,
     1          ' 4.Y AXIS ANGLE = ',F10.3,/,
     1          ' 5.Z AXIS ANGLE = ',F10.3,/,
     1          ' 6.X MIN = ',F10.3,/,
     1          ' 7.X MAX = ',F10.3,/,
     1          ' 8.Y MIN = ',F10.3,/,
     1          ' 9.Y MAX = ',F10.3)
 106    FORMAT(' 10. Z VARIABLE = ',I2,'. ',10A1)
 107    FORMAT(' ENTER NUMBER OF PARAMETER TO CHANGE AND NEW',
     1          ' VALUE (IF APPROPRIATE)')
 108    FORMAT(I3,F10.0)
 109    FORMAT(' ENTER NEW TITLE, 60 CHARACTERS MAX')
 110    FORMAT(60A1)
 111    FORMAT(' 11.ZMIN = ',F10.5,/,
     1          ' 12.ZMAX = ',F10.5,/,
     1          ' 13.ZINC = ',F10.5)
 112    FORMAT(' DO YOU WANT A HARDCOPY PLOT?')
 113    FORMAT(A1)
 114    FORMAT(' 1.OPTION = ',F2.0,/,
     1          ' 2.TITLE = ',60A1,/,
     1          ' 3.X AXIS ANGLE = ',F10.3,/,
     1          ' 4.Y AXIS ANGLE = ',F10.3,/,
     1          ' 5.Z AXIS ANGLE = ',F10.3,/,
     1          ' 6.X MIN = ',F10.3,/,
     1          ' 7.X MAX = ',F10.3,/,
    1       ' 8.Y MIN = ',F10.3,/,
    1       ' 9.Y MAX = ',F10.3,/,
    1       ' 10.VECTOR LENGTH SCALE FACTOR = ',F10.3)
 115  FORMAT(' DATA XMIN - ',F10.3,/,
    1       ' DATA XMAX = ',F10.3,/,
    1       ' DATA YMIN = ',F10.3,/,
    1       ' DATA YMAX = ',F10.3)
 116  FORMAT(' DATA ZMIN = ',F10.3,/,
    1       'D DATA ZMAX = ',F10.3)
 117  FORMAT(' 14.Z SCALE FACTOR - ',F10.3)
 118  FORMAT(' DO YOU WANT TO SEE THE DATA MINIMUM/MAXIMUM',
    1       ' VALUES? [Y/N]')
 119  FORMAT(A1)
C**************************************************
C
C
C     DATA PRESETS AREA
C
C
C**************************************************
      DATA YES/'Y'/
      DATA MAXSIZ/100/,ND/5/,LUN/3/
C**************************************************
C
C
C     MAIN PROGRAM CODE STARTS HERE
C
C
C**************************************************
      CALL INITT(960)
      DEFALT=.FALSE.
      NOCHG=.TRUE.
      MNMX=.FALSE.
      EOF=.FALSE.
      CALL ASSIGN(2,'MENU.DAT',8)
      CALL FDBSET(2,'READONLY')
      READ(2,END=200)TITLE,OPTION,HRDCPY,A1,A2,A3,A4,S1,S2,S3,S4,
    1       XMIN,XMAX,XINC,YMIN,YMAX,YINC,ZMIN,ZMAX,ZINC,IDLIST
      IOPT=INT(OPTION)
      WRITE(5,118)
      READ(5,119)ANS
      IF (ANS.NE.YES)MNMX=.TRUE.
      CALL CLOSE(2)
      IF (MNMX)GO TO 1000
      GO TO 610
C     EOF WAS ENCOUNTERED ON MENU FILE - GET DESIRED OPTION
C     AND USE THE APPROPRIATE DEFAULTS FOR THAT OPTION
 200  CONTINUE
      DEFALT=.TRUE.
      IDLIST(1)=2
      IDLIST(2)=1
      WRITE(5,101)
      READ(5,102)OPTION
      IOPT=INT(OPTION)
      GO TO (300,400,500)IOPT+1
      GO TO 200
C     CONTOUR PLOT OPTION SELECTED
 300  CONTINUE
      A1=0.
      A2=90.
      A3=0.
      S1=1.
      S2=1.
      S3=0.
      A4=0.
      S4=0.
      IDLIST(3)=10
      IDLIST(4)=3
      IDLIST(5)=11
      GO TO 600
C     AXONOMETRIC PROJECTION PLOT SELECTED
 400  CONTINUE
      A1=0.
      A2=120.
      A3=90.
      A4=0.
      S1=1.
      S2=1.
      S3=-1.
      S4=0.
      IDLIST(3)=10
      IDLIST(4)=3
      IDLIST(5)=11
      GO TO 600
C     CROSS FLOW PLOT SELECTED
 500  CONTINUE
      A1=0.
      A2=90.
      A3=0.
      A4=0.
      S1=1.
      S2=1.
      S3=.5
      S4=0.
      IDLIST(3)=19
      IDLIST(4)=20
      IDLIST(5)=3
C     GET DATA MINIMUM AND MAXIMUMS FROM THE DATA
C     FILE AND DETERMINE THE DATA INCREMENTS NECESSARY
C     TO PLOT THE DATA
 600  CONTINUE
      CALL CLOSE(2)
 610  CONTINUE
      CALL ASSIGN(3,'DATAIN.DAT',10)
      CALL FDBSET(3,'OLD','SHARE')
      DXMIN=999.
      DXMAX=-999.
      DYMIN=999.
      DYMAX=-999.
      DZMIN=999.
      DZMAX=-999.
 700  CONTINUE
      CALL GETGRD(LUN,ND,IDLIST,MAXSIZ,LSCR,X,STITLE,TITLE,DLAB,
     1            DAT,NF,NL,NG,NLAT,EOF)
      IF (EOF)GO TO 800
      CALL MINMAX(DAT,DXMIN,DXMAX,DYMIN,DYMAX,DZMIN,DZMAX)
      GO TO 700
 800  CONTINUE
      CALL CLOSE(3)
      IF (DEFALT.EQ..FALSE.)GO TO 1000
      XMAX=DXMAX
      YMAX=DYMAX
      ZMAX=DZMAX
      XMIN=DXMIN
      YMIN=DYMIN
      ZMIN=DZMIN
      XM=XMIN
      YM=YMIN
      ZM=ZMIN
      CALL INCRMT(XM,XMAX,XINC,5.)
      CALL INCRMT(YM,YMAX,YINC,5.)
      CALL INCRMT(ZM,ZMAX,ZINC,10.)
      XINC=AMAX1(XINC,YINC)
      YINC=XINC
      XMIN=XINC*AINT(XM/XINC)
      YMIN=YINC*AINT(YM/YINC)
      ZMIN=ZINC*AINT(ZM/ZINC)
      IF (XM.LT.0.)XMIN=XMIN-XINC
      IF (YM.LT.0.)YMIN=YMIN-YINC
      XMAX=XMIN+5.*XINC
      YMAX=YMIN+5.*YINC
      IF (ZMIN.LT.ZM)ZMIN=ZMIN+ZINC
C****************************************************
C
C
C     DISPLAY PARAMETERS AND ALLOW CHANGES
C
C
C******************************************************
 1000 CONTINUE
      IF (OPTION.GT.1.)GO TO 2000
      CALL NEWPAG
      IF (MNMX)GO TO 1005
      WRITE(5,115)DXMIN,DXMAX,DYMIN,DYMAX
      WRITE(5,116)DZMIN,DZMAX
 1005 CONTINUE
      WRITE(5,105)OPTION,TITLE,A1,A2,A3,XMIN,XMAX,YMIN,YMAX
      WRITE(5,106)IDLIST(3),(DLAB(I,3),I=1,10)
      IF (OPTION.LT.1.)WRITE(5,111)ZMIN,ZMAX,ZINC
      IF (OPTION.EQ.1.)WRITE(5,117)S3
      WRITE(5,107)
      READ(5,108)IPARAM,VALU
      IF (IPARAM.EQ.0)GO TO 3000
      GO TO (1010,1020,1030,1040,1050,1060,1070,1080,1150)IPARAM
      GO TO (1090,1120,1130,1140,1160)IPARAM-9
      GO TO 1000
 1010 CONTINUE
      IOPT=INT(VALU)
      IF ((IOPT.LT.0).OR.(IOPT.GT.2))GO TO 1000
      OPTION=IOPT
      GO TO (300,400,500)IOPT+1
 1020 WRITE(5,109)
      READ(5,110)TITLE
      GO TO 1000
 1030 A1=VALU
      GO TO 1000
 1040 A2=VALU
      GO TO 1000
 1050 A3=VALU
      GO TO 1000
 1060 XMIN=VALU
      NOCHG=.FALSE.
      GO TO 1000
```

```
1070    XMAX=VALU
        NOCHG=.FALSE.
        GO TO 1000
1080    YMIN=VALU
        NOCHG=.FALSE.
        GO TO 1000
1090    CONTINUE
        IV=INT(VALU)
        IF ((IV.GT.0).AND.(IV.LT.21))GO TO 1095
        WRITE(5,103)
        WRITE(5,104)DLABEL
        GO TO 1000
1095    CONTINUE
        IDLIST(3)=IV
        GO TO 610
1120    ZMIN=VALU
        GO TO 1000
1130    ZMAX=VALU
        GO TO 1000
1140    ZINC=VALU
        GO TO 1000
1150    CONTINUE
        YMAX=VALU
        NOCHG=.FALSE.
        GO TO 1000
1160    CONTINUE
        S3=VALU
        GO TO 1000
2000    CONTINUE
        CALL NEWPAG
        IF (MNMX)GO TO 2005
        WRITE(5,115)DXMIN,DXMAX,DYMIN,DYMAX
2005    CONTINUE
        WRITE(5,114)OPTION,TITLE,A1,A2,A3,XMIN,XMAX,YMIN,YMAX,S3
        WRITE(5,107)
        READ(5,108)IPARAM,VALU
        IF (IPARAM.EQ.0)GO TO 2000
        GO TO (2010,2020,2030,2040,2050,2060,2070,2080,2090)IPARAM
        IF (IPARAM.EQ.10)GO TO 2095
        GO TO 2000
2010    CONTINUE
        IOPT=INT(VALU)
        IF ((IOPT.LT.0).OR.(IOPT.GT.2))GO TO 2000
        OPTION=IOPT
        GO TO (300,400,500)IOPT+1
2020    WRITE(5,109)
        READ(5,110)TITLE
        GO TO 2000
2030    A1=VALU
        GO TO 2000
2040    A2=VALU
        GO TO 2000
2050    A3=VALU
        GO TO 2000
2060    XMIN=VALU
        NOCHG=.FALSE.
        GO TO 2000
2070    XMAX=VALU
        NOCHG=.FALSE.
        GO TO 2000
2080    YMIN=VALU
```

```
        NOCHG=.FALSE.
        GO TO 2000
 2090   CONTINUE
        YMAX=VALU
        NOCHG=.FALSE.
        GO TO 2000
 2095   CONTINUE
        S3=VALU
        GO TO 2000
C*****************************************************************
C
C
C       CLOSE FILES AND WRITE NEW MENU.DAT FILE TO DISK
C
C
C*******************************************************************
 3000   CONTINUE
        IF (NOCHG)GO TO 3100
        XM=XMIN
        YM=YMIN
        CALL INCRMT(XM,XMAX,XINC,5.)
        CALL INCRMT(YM,YMAX,YINC,5.)
        XINC=AMAX1(XINC,YINC)
        YINC=XINC
        XMIN=XINC*AINT(XM/XINC)
        YMIN=YINC*AINT(YM/YINC)
        IF (XM.LT.0.)XMIN=XMIN-XINC
        IF (YM.LT.0.)YMIN=YMIN-YINC
        XMAX=XMIN+5.*XINC
        YMAX=YMIN+5.*YINC
 3100   CONTINUE
        CALL ASSIGN(2,'MENU.DAT','8)
        CALL FDBSET(2,'NEW','SHARE')
        WRITE(5,112)
        READ(5,113)HRDCPY
        WRITE(2)TITLE,OPTION,HRDCPY,A1,A2,A3,A4,S1,S2,S3,S4,
     1          XMIN,XMAX,XINC,YMIN,YMAX,YINC,ZMIN,ZMAX,ZINC,IDLIST
        CALL CLOSE(2)
        CALL FINITT(0,100)
        END
C******************************************************************
C
C
C       THIS ROUTINE COMPUTES THE STANDARD INCREMENT FOR EACH AXIS
C
C
C*********************************************************************
        SUBROUTINE INCRMT(XMIN,XMAX,XINC,COUNT)
        DIMENSION FRAC(4)
        DATA FRAC/0.,.3010299956,.6989700041,1./
        XINC=(XMAX-XMIN)/COUNT
        A=ALOG10(XINC)
        WHOLE=AINT(A)
        FRACT=ABS(A-WHOLE)
        IX=1
        IF (FRACT.GT.FRAC(2))IX=2
        IF (FRACT.GT.FRAC(3))IX=3
        IF (A.GT.0)IX=IX+1
        A=WHOLE+SIGN(FRAC(IX),A)
        XINC=10.**A
        RETURN
```

```
      END
C***********************************************************************
C
C
C       THIS ROUTINE FINDS THE MINIMUM AND MAXIMUM VALUES
C       IN ARRAY DAT FOR X, Y, AND Z.
C
C
C***********************************************************************
      SUBROUTINE MINMAX(DAT,XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX)
      DIMENSION DAT(5,100)
      DO 100 I=1,100
      XMIN=AMIN1(XMIN,DAT(1,I))
      YMIN=AMIN1(YMIN,DAT(2,I))
      ZMIN=AMIN1(ZMIN,DAT(3,I))
      XMAX=AMAX1(XMAX,DAT(1,I))
      YMAX=AMAX1(YMAX,DAT(2,I))
      ZMAX=AMAX1(ZMAX,DAT(3,I))
 100  CONTINUE
      RETURN
      END
```

```
0123456789012345G789    ** RSX-11M V3.2 **    18-SEP-81    14:23:29    DF0:[152,13]GSUBS.FTN;43    01234567890123456789
01234567890123456789    ** RSX-11M V3.2 **    18-SEP-81    14:23:29    DF0:[152,13]GSUBS.FTN;43    01234567890123456789
01234567890123456789    ** RSX-11M V3.2 **    18-SEP-81    14:23:29    DF0:[152,13]GSUBS.FTN;43    01234567890123456789



  GGGGGGGG     SSSSSSSS   UU      UU   BBBBBBBB       SSSSSSSS
  GGGGGGGG     SSSSSSSS   UU      UU   BBBBBBBB       SSSSSSSS
GG           SS           UU      UU   BB      BB   SS
GG           SS           UU      UU   BB      BB   SS
GG           SS           UU      UU   BB      BB   SS
GG           SS           UU      UU   BB      BB   SS
GG             SSSSSS     UU      UU   BBBBBBBB       SSSSSS
GG             SSSSSS     UU      UU   BBBBBBBB       SSSSSS
GG  GGGGGG           SS   UU      UU   BB      BB           SS
GG  GGGGGG           SS   UU      UU   BB      BB           SS
GG      GG           SS   UU      UU   BB      BB           SS        ....
GG      GG           SS   UU      UU   BB      BB           SS        ....
  GGGGGG     SSSSSSSS     UUUUUUUUUU   BBBBBBBB     SSSSSSSS          ....
  GGGGGG     SSSSSSSS     UUUUUUUUUU   BBBBBBBB     SSSSSSSS          ....



FFFFFFFFFF   TTTTTTTTTT   NN      NN        ;;;;     44    44       333333
FFFFFFFFFF   TTTTTTTTTT   NN      NN        ;;;;     44    44       333333
FF               TT       NN      NN        ;;;;     44    44     33      33
FF               TT       NN      NN        ;;;;     44    44     33      33
FF               TT       NNNN    NN                 44    44             33
FF               TT       NNNN    NN                 44    44             33
FFFFFFFF         TT       NN  NN  NN        ;;;;     4444444444         33
FFFFFFFF         TT       NN  NN  NN        ;;;;     4444444444         33
FF               TT       NN    NNNN        ;;;;           44             33
FF               TT       NN    NNNN        ;;;;           44             33
FF               TT       NN      NN          ;;           44     33      33
FF               TT       NN      NN          ;;           44     33      33
FF               TT       NN      NN        ;;             44       333333
FF               TT       NN      NN        ;;             44       333333



01234567890123456789    ** RSX-11M V3.2 **    18-SEP-81    14:23:29    DF0:[152,13]GSUBS.FTN;43    01234567890123456789
01234567890123456789    ** RSX-11M V3.2 **    18-SEP-81    14:23:29    DF0:[152,13]GSUBS.FTN;43    01234567890123456789
01234567890123456789    ** RSX-11M V3.2 **    18-SEP-81    14:23:29    DF0:[152,13]GSUBS.FTN;43    01234567890123456789
```

```
C*****************************************************
C
C
C       THIS ROUTINE CHECKS TO SEE IF X AND Y ARE
C       WITHIN THE LIMITS DEFINED.  IF NOT IT
C       SET THE VISIBILITY FLAG TO INVISIBLE.
C
C
C*****************************************************
        SUBROUTINE LIMITS(DAT,NTOT,XMIN,XMAX,YMIN,YMAX,IX,ZMIN,OPT)
        DIMENSION DAT(5,100)
        DO 100 I=1,NTOT
        IF ((DAT(1,I).LT.XMIN).OR.(DAT(1,I).GT.XMAX))DAT(IX,I)=1.
        IF ((DAT(2,I).LT.YMIN).OR.(DAT(2,I).GT.YMAX))DAT(IX,I)=1.
        DAT(1,I)=DAT(1,I)-XMIN
        DAT(2,I)=DAT(2,I)-YMIN
        IF (OPT.EQ.1.)DAT(3,I)=DAT(3,I)-ZMIN
 100    CONTINUE
        RETURN
        END
C********************************************************
C
C
C       THIS SUBROUTINE PLOTS A SINGLE CONTOUR LINE
C
C
C**********************************************************
        SUBROUTINE CONTOR(N,X1,Y1,Z1,V1,X2,Y2,Z2,V2,X3,Y3,Z3,LINE)
C
C
C    INPUT:
C     N       NO. OF VALUES IN COORDINATE ARRAYS
C     X1,Y1,Z1  INPUT COORDINATES OF ONE RAIL
C     V1      VISIBILITY INDICATOR OF ONE RAIL
C     X2,Y2,Z2  INPUT COORDINATES OF SECOND RAIL
C     V2      VISIBILITY INDICATOR OF SECOND RAIL
C     X3,Y3   SCRATCH ARRAYS
C     Z3      Z VALUE OF CONTOUR LINE
C     LINE    LINE STYLE PARAMETER FOR CONTOUR LINE
C
C
        DIMENSION X1(N),Y1(N),Z1(N),V1(N),X2(N),Y2(N),Z2(N),V2(N),X3(N),
     1   Y3(N)
        LOGICAL FIRST
        FIRST=.FALSE.
        K=1
        N1=1
        N2=1
        IS=1
        IF((Z1(1)-Z3)*(Z2(1)-Z3).GT.0.)GO TO 10
        CALL INTER(X1,Y1,Z1,V1,X2,Y2,Z2,V2,X3,Y3,Z3)
        K=K+1
        FIRST=.TRUE.
10      CONTINUE
        CALL CHANGE(N1,N,Z1,Z3)
        CALL CHANGE(N2,N,Z2,Z3)
        IF(FIRST)GO TO 40
20      CONTINUE
        IS=MIN0(N1,N2)
        IF(IS.EQ.999)RETURN
```

```
      IS1=IS-1
      IF(IS.EQ.N1)GO TO 30
      CALL INTER(X2(IS1),Y2(IS1),Z2(IS1),V2(IS1),X2(IS),Y2(IS),Z2(IS),
     *  V2(IS),X3(K),Y3(K),Z3)
      K=K+1
      CALL CHANGE(N2,N,Z2,Z3)
      GO TO 40
30    CONTINUE
      CALL INTER(X1(IS1),Y1(IS1),Z1(IS1),V1(IS1),X1(IS),Y1(IS),Z1(IS),
     *  V1(IS),X3(K),Y3(K),Z3)
      K=K+1
      CALL CHANGE(N1,N,Z1,Z3)
40    CONTINUE
      IF=MIN0(N1,N2)
      IF1=IF-1
      IF(IS.GT.IF1)GO TO 55
      DO 50 I=IS,IF1
      IF(I.GT.N)GO TO 70
      CALL INTER(X1(I),Y1(I),Z1(I),V1(I),X2(I),Y2(I),Z2(I),V2(I),X3(K),
     1 Y3(K),Z3)
      K=K+1
50    CONTINUE
      IF(IF.EQ.N1)GO TO 60
55    CONTINUE
      CALL INTER(X2(IF1),Y2(IF1),Z2(IF1),V2(IF1),X2(IF),Y2(IF),Z2(IF),
     *  V2(IF),X3(K),Y3(K),Z3)
      CALL CHANGE(N2,N,Z2,Z3)
      GO TO 80
60    CONTINUE
      CALL INTER(X1(IF1),Y1(IF1),Z1(IF1),V1(IF1),X1(IF),Y1(IF),Z1(IF),
     *  V1(IF),X3(K),Y3(K),Z3)
      CALL CHANGE(N1,N,Z1,Z3)
      GO TO 80
70    K=K-1
80    CONTINUE
      ISTART=1
      DO 90 LLL=1,K
      KK=LLL
      H=X3(KK)
      V=Y3(KK)
      IF((ISTART.EQ.1).AND.(X3(KK).NE.-999.))CALL DASHA(H,V,-1)
      IF((ISTART.EQ.0).AND.(X3(KK).NE.-999.))CALL DASHA(H,V,LINE)
      ISTART=0
      IF(X3(KK).EQ.-999.)ISTART=1
90    CONTINUE
      K=1
      GO TO 20
      END
C****************************************************************
C
C
C    THIS ROUTINE LINEARILY INTERPOLATES BETWEEN THE INPUT POINTS
C    FOR THE POINT WITH THE SPECIFIED Z VALUE
C
C
C*******************************************************************
      SUBROUTINE INTER(X1,Y1,Z1,V1,X2,Y2,Z2,V2,X3,Y3,Z3)
C
C
C    INPUT:
C      X1,Y1,Z1  INPUT COORDINATES OF THE FIRST POINT
```

```
C         V1            VISIBILITY PARAMETER OF THE FIRST POINT
C         X2,Y2,Z2      INPUT COORDINATES OF THE SECOND POINT
C         V2            VISIBILITY PARAMETER OF THE SECOND POINT
C         Z3            SPECIFIED Z VALUE
C       OUTPUT:
C         X3,Y3,Z3      OUTPUT COORDINATES OF THE POINT
C
          P=1
          IF(Z2.EQ.Z3)GO TO 10
          P=(Z3-Z1)/(Z2-Z1)
10        CONTINUE
          X3=X1+P*(X2-X1)
          Y3=Y1+P*(Y2-Y1)
          IF((V1.NE.0.).OR.(V2.NE.0.))X3=-999.
          RETURN
          END
C*****************************************************************
C
C
C       THIS ROUTINE DETERMINES WHERE THE DATA IN THE Z1 ARRAY CROSSES
C       THE Z3 VALUE
C
C
C*******************************************************************
          SUBROUTINE CHANGE(N1,N,Z1,Z3)
C
C
C       INPUT:
C         N1        LOCATION IN ARRAY OF THE PREVIOUS CROSSING
C         N         TOTAL NO. OF VALUES IN THE Z1 ARRAY
C         Z1        INPUT ARRAY
C         Z3        CROSSING VALUE
C       OUTPUT:
C         N         LOCATION IN ARRAY OF THE NEXT CROSSING
C
          DIMENSION Z1(N)
          IF(N1.GE.N)GO TO 20
          N11=N1+1
          DO 10 I=N11,N
          N1=I
          IF((Z1(I-1)-Z3)*(Z1(I)-Z3).LE.0.)RETURN
10        CONTINUE
20        N1=999
          RETURN
          END
C*******************************************************************
C
C
C       THIS ROUTINE DISPLAYS A LEGEND DEFINING WHAT VALUES THE DIFFERENT
C       LINE STYLES REPRESENT
C
C
C********************************************************************
          SUBROUTINE LEGEND(N,LINE,ZLOW,ZINC,XS,YS,XINC,YINC)
C
C
C       INPUT:
C         N           NO. OF LINE STYLES TO DISPLAY
C         LINE        LINE STYLE PARAMETER ARRAY
C         ZLOW        Z VALUE FIRST LINE STYLE REPRESENTS
C         ZINC        Z INCREMENT BETWEEN LINES
```

```
C       XS,YS     COORDINATES OF LEFT SIDE OF FIRST LINE
C       XINC      COORDINATE OF LEFT SIDE OF SIXTH LINE IF NECESSARY
C       YINC      COORDINATE DELTA BETWEEN LINES
C
         DIMENSION LINE(N)
         LOGICAL*1 NADE(10)
         Z3=ZLOW
         Y=YS
         X=XS
         ICNT=0
         DO 30 I=1,N
         CALL DASHA(X,Y,-1)
         CALL DASHA(X+XINC/2.,Y,LINE(I))
         ENCODE(10,99,NADE)Z3
         NADE(1)=32
         CALL AANSTR(10,NADE)
         Z3=Z3+ZINC
         Y=Y-YINC
         ICNT=ICNT+1
 30      CONTINUE
         RETURN
99       FORMAT(F10.3)
         END
C**********************************************************************
C
C
C    THIS ROUTINE DISPLAYS A GRID BY DRAWING THE RUNGS FIRST AND THEN
C    THE SECOND RAIL
C
C
C***********************************************************************
         SUBROUTINE D3MESH(F,N,X1,Y1,Z1,W1,X2,Y2,Z2,W2)
C
C
C    INPUT:
C      F          PROJECTION MATRIX
C      N          NO. OF VALUES IN COORDINATE ARRAYS
C      X1,Y1,Z1     COORDINATES OF FIRST RAIL
C      W1         VISIBILITY PARAMETERS FOR FIRST RAIL
C      X2,Y2,Z2     COORDINATES OF SECOND RAIL
C      W2         VISIBILTIY PARAMETERS FOR SECOND RAIL
C
         DIMENSION F(4,2),X1(N),Y1(N),Z1(N),W1(N),X2(N),Y2(N),Z2(N),W2(N)
         H1(I)=F(1,1)*X1(I)+F(2,1)*Y1(I)+F(3,1)*Z1(I)+F(4,1)
         V1(I)=F(1,2)*X1(I)+F(2,2)*Y1(I)+F(3,2)*Z1(I)+F(4,2)
         H2(I)=F(1,1)*X2(I)+F(2,1)*Y2(I)+F(3,1)*Z2(I)+F(4,1)
         V2(I)=F(1,2)*X2(I)+F(2,2)*Y2(I)+F(3,2)*Z2(I)+F(4,2)
         DO 50 LLL=1,N
         I=LLL
         IF((W1(I).NE.0.).OR.(W2(I).NE.0.))GO TO 50
         H=H1(I)
         V=V1(I)
         CALL MOVEA(H,V)
         H=H2(I)
         V=V2(I)
         CALL DRAWA(H,V)
50       CONTINUE
         ISTART=1
         DO 75 LLL=1,N
         I=LLL
         H=H2(I)
```

```
          V=V2(I)
          IF((ISTART.EQ.1).AND.(W2(I).EQ.0.))CALL MOVEA(H,V)
          IF((ISTART.EQ.0).AND.(W2(I).EQ.0.))CALL DRAWA(H,V)
          ISTART=0
          IF(W2(I).NE.0.)ISTART=1
75        CONTINUE
          RETURN
          END
C**********************************************************************
C
C
C     THIS ROUTINE DEFINES AN AXONOMETRIC PROJECTION MATRIX
C
C
C***********************************************************************
          SUBROUTINE FFILL(F,XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX,
     1             SXMIN,SXMAX,SYMIN,SYMAX)
C
C
C     INPUT:
C       F             INCLUDES 3 ANGLES AXES MAKE WITH THE HORIZONTAL AND 3
C                     SCALE FACTORS
C       XMIN,XMAX  LIMITS OF COORDINATE DATA THAT WILL BE PLOTTED
C       YMIN,YMAX  SAME AS ABOVE
C       ZMIN,ZMAX  SAME AS ABOVE
C        SXMIN,SYMIN      MINIMUM SCALE HORIZONTAL AND VERTICAL VALUES
C        SXMAX,SYMAX      MAXIMUM SCALE HORIZONTAL AND VERTICAL VALUES
C
          DIMENSION F(4,2),A(3)
          H(X,Y,Z)=X*F(1,1)+Y*F(2,1)+Z*F(3,1)+F(4,1)
          V(X,Y,Z)=X*F(1,2)+Y*F(2,2)+Z*F(3,2)+F(4,2)
          DO 5 I=1,3
          A(I)=F(I,1)/57.296
          F(I,1)=COS(A(I))*F(I,2)
          F(I,2)=SIN(A(I))*F(I,2)
5         CONTINUE
          H1=H(XMIN,YMIN,ZMIN)
          V1=V(XMIN,YMIN,ZMIN)
          H2=H(XMIN,YMIN,ZMAX)
          V2=V(XMIN,YMIN,ZMAX)
          H3=H(XMIN,YMAX,ZMIN)
          V3=V(XMIN,YMAX,ZMIN)
          H4=H(XMAX,YMIN,ZMIN)
          V4=V(XMAX,YMIN,ZMIN)
          H5=H(XMIN,YMAX,ZMAX)
          V5=V(XMIN,YMAX,ZMAX)
          H6=H(XMAX,YMAX,ZMIN)
          V6=V(XMAX,YMAX,ZMIN)
          H7=H(XMAX,YMIN,ZMAX)
          V7=V(XMAX,YMIN,ZMAX)
          H8=H(XMAX,YMAX,ZMAX)
          V8=V(XMAX,YMAX,ZMAX)
          SXMIN=AMIN1(H1,H2,H3,H4,H5,H6,H7,H8)
          SXMAX=AMAX1(H1,H2,H3,H4,H5,H6,H7,H8)
          SYMIN=AMIN1(V1,V2,V3,V4,V5,V6,V7,V8)
          SYMAX=AMAX1(V1,V2,V3,V4,V5,V6,V7,V8)
          RETURN
          END
C***********************************************************************
C
C
```

```
C       THIS ROUTINE DRAWS AND LABELS A GRID SPACE
C
C
C********************************************************************
        SUBROUTINE GRID(F,XMIN,XMAX,XINC,YMIN,YMAX,YINC)
        DIMENSION F(4,2)
        LOGICAL*1 NADE(10)
        H(X,Y)=X*F(1,1)+Y*F(2,1)
        V(X,Y)=X*F(1,2)+Y*F(2,2)
        HH=H(XMIN,YMIN)
        VV=V(XMIN,YMIN)
        CALL MOVEA(HH,VV)
        HH=H(XMAX,YMIN)
        VV=V(XMAX,YMIN)
        CALL DRAWA(HH,VV)
        HH=H(XMAX,YMAX)
        VV=V(XMAX,YMAX)
        CALL DRAWA(HH,VV)
        HH=H(XMIN,YMAX)
        VV=V(XMIN,YMAX)
        CALL DRAWA(HH,VV)
        HH=H(XMIN,YMIN)
        VV=V(XMIN,YMIN)
        CALL DRAWA(HH,VV)
        X=XMIN
        Y=YMIN-.015*(YMAX-YMIN)
        DO 20 I=1,6
        HH=H(X,Y)
        VV=V(X,Y)
        CALL MOVEA(HH,VV)
        HH=H(X,YMIN)
        VV=V(X,YMIN)
        CALL DRAWA(HH,VV)
        X=X+.2*(XMAX-XMIN)
 20     CONTINUE
        Y=YMIN
        X=XMAX+.015*(XMAX-XMIN)
        DO 30 I=1,6
        HH=H(X,Y)
        VV=V(X,Y)
        CALL MOVEA(HH,VV)
        HH=H(XMAX,Y)
        VV=V(XMAX,Y)
        CALL DRAWA(HH,VV)
        Y=Y+.2*(YMAX-YMIN)
 30     CONTINUE
        ENCODE(10,99,NADE)XMIN
        NADE(1)=32
        X=XMIN-.045*(XMAX-XMIN)
        Y=YMIN-.05*(YMAX-YMIN)
        HH=H(X,Y)
        VV=V(X,Y)
        CALL MOVEA(HH,VV)
        CALL AANSTR(10,NADE)
        ENCODE(10,99,NADE)XMAX
        NADE(1)=32
        X=XMAX-.05*(XMAX-XMIN)
        HH=H(X,Y)
        VV=V(X,Y)
        CALL MOVEA(HH,VV)
        CALL AANSTR(10,NADE)
```

```
      ENCODE(10,99,NADE)YMIN
      NADE(1)=32
      X=XMAX+.05*(XMAX-XMIN)
      HH=H(X,YMIN)
      VV=V(X,YMIN)
      CALL MOVEA(HH,VV)
      CALL AANSTR(10,NADE)
      ENCODE(10,99,NADE)YMAX
      NADE(1)=32
      HH=H(X,YMAX)
      VV=V(X,YMAX)
      CALL MOVEA(HH,VV)
      CALL AANSTR(10,NADE)
 99   FORMAT(F10.3)
      RETURN
      END
C####################################################
C
C
C     THIS ROUTINE PLOTS A VECTOR FIELD
C
C
C#####################################################
      SUBROUTINE VFIELD(N,DAT,SCALEF)
      DIMENSION DAT(5,100)
      DO 10 I=1,N
      IF(DAT(5,I).NE.0.)GO TO 10
      CALL MOVEA(DAT(1,I),DAT(2,I))
      UU=DAT(3,I)*SCALEF
      VV=DAT(4,I)*SCALEF
      CALL ARROW(-UU,VV)
 10   CONTINUE
      RETURN
      END
C#####################################################
C
C
C     THIS ROUTINE DRAWS THE ARROWS FOR A VECTOR FIELD PLOT
C
C
C#####################################################
      SUBROUTINE ARROW(U,V)
      DATA S2,C2/.707,-.707/
      CALL DRAWR(U,V)
      W=SQRT(U*U+V*V)
      IF(W.LE..01)RETURN
      S1=V/W
      C1=U/W
      UU=W/5.*(C1*C2-S1*S2)
      VV=W/5.*(S1*C2+C1*S2)
      CALL DRAWR(UU,VV)
      RETURN
      END
C##################################################
C
C
C     THIS ROUTINE OUTPUTS AN ALPHANUMERIC STRING TO THE SCREEN
C
C
C####################################################
      SUBROUTINE AANSTR(N,STRING)
```

```
        LOGICAL*1 STRING(N)
        DIMENSION NAD(60)
        DO 10 I=1,N
 10     NAD(I)=STRING(I)
        CALL ANSTR(N,NAD)
        RETURN
        END
C******************************************************************
C
C
C       THIS ROUTINE FILLS ARRAYS USED FOR PLOTTING FROM THE
C       DAT ARRAY
C
C
C*******************************************************************
        SUBROUTINE FILLIT(NP,K1,K2,X1,Y1,Z1,V1,X2,Y2,Z2,V2,DAT)
        DIMENSION X1(NP),Y1(NP),Z1(NP),V1(NP),X2(NP),Y2(NP),
     1          Z2(NP),V2(NP),DAT(5,100)
        DO 115 JJ=1,NP
        X1(JJ)=DAT(1,K1+JJ-1)
        Y1(JJ)=DAT(2,K1+JJ-1)
        Z1(JJ)=DAT(3,K1+JJ-1)
        V1(JJ)=DAT(4,K1+JJ-1)
        X2(JJ)=DAT(1,K2+JJ-1)
        Y2(JJ)=DAT(2,K2+JJ-1)
        Z2(JJ)=DAT(3,K2+JJ-1)
        V2(JJ)=DAT(4,K2+JJ-1)
 115    CONTINUE
        RETURN
        END
```

```
0123456789012345G789    ** RSX-11M V3.2 **     18-SEP-81    14:25:26    DP0:[152,13]COMCAL.FTN;20    01234567890123456789
01234567890123456789    ** RSX-11M V3.2 **     18-SEP-81    14:25:26    DP0:[152,13]COMCAL.FTN;20    01234567890123456789
01234567890123456789    ** RSX-11M V3.2 **     18-SEP-81    14:25:26    DP0:[152,13]COMCAL.FTN;20    01234567890123456789




  CCCCCCCC     000000     MM       MM     CCCCCCCC     AAAAAA     LL
  CCCCCCCC     000000     MM       MM     CCCCCCCC     AAAAAA     LL
CC           00      00   MMMM   MMMM   CC           AA      AA   LL
CC           00      00   MMMM   MMMM   CC           AA      AA   LL
CC           00      00   MM  MM   MM   CC           AA      AA   LL
CC           00      00   MM  MM   MM   CC           AA      AA   LL
CC           00      00   MM       MM   CC           AA      AA   LL
CC           00      00   MM       MM   CC           AA      AA   LL
CC           00      00   MM       MM   CC           AAAAAAAAAA   LL
CC           00      00   MM       MM   CC           AAAAAAAAAA   LL
CC           00      00   MM       MM   CC           AA      AA   LL            ....
CC           00      00   MM       MM   CC           AA      AA   LL            ....
  CCCCCCCC     000000     MM       MM     CCCCCCCC   AA      AA   LLLLLLLLLL    ....
  CCCCCCCC     000000     MM       MM     CCCCCCCC   AA      AA   LLLLLLLLLL    ....




FFFFFFFFFF   TTTTTTTTTT   NN       NN      ;;;;       222222       000000
FFFFFFFFFF   TTTTTTTTTT   NN       NN      ;;;;       222222       000000
FF               TT       NN       NN      ;;;;     22      22   00      00
FF               TT       NN       NN      ;;;;     22      22   00      00
FF               TT       NNNN     NN                       22   00    0000
FF               TT       NNNN     NN                       22   00    0000
FFFFFFFF         TT       NN  NN   NN      ;;;;           22     00  00  00
FFFFFFFF         TT       NN  NN   NN      ;;;;           22     00  00  00
FF               TT       NN    NNNN       ;;;;         22       0000    00
FF               TT       NN    NNNN       ;;;;         22       0000    00
FF               TT       NN       NN        ;;       22         00      00
FF               TT       NN       NN        ;;       22         00      00
FF               TT       NN       NN      ;;       2222222222     000000
FF               TT       NN       NN      ;;       2222222222     000000




01234567890123456789    ** RSX-11M V3.2 **     18-SEP-81    14:25:26    DP0:[152,13]COMCAL.FTN;20    01234567890123456789
01234567890123456789    ** RSX-11M V3.2 **     18-SEP-81    14:25:26    DP0:[152,13]COMCAL.FTN;20    01234567890123456789
01234567890123456789    ** RSX-11M V3.2 **     18-SEP-81    14:25:26    DP0:[152,13]COMCAL.FTN;20    01234567890123456789
```

```
C
C
C        THIS SUBROUTINE IS USED TO CONVERT DATA TAKEN WITH
C        A SEVEN HOLE PROBE INTO USEFUL QUANTITIES.  THIS
C        PROGRAM IS WRITTEN TO BE USED WITH A PROBE THAT
C        HAS BEEN CALIBRATED FOR HIGH SPEED COMPRESSIBLE
C        FLOW.  THE CALIBRATION COEFFICIENTS ARE STORED IN
C        A FILE NAMED 'CPROBE.DAT'.  IT IS THE USERS JOB
C        TO ACCESS THIS FILE AND PASS THE CALIBRATION
C        COEFFICIENTS TO THE SUBROUTINE.  THE FILE MAY BE
C        ACCESSED AS FOLLOWS:
C
C
C        CALL ASSIGN(2,'CPROBE.DAT',10)
C        CALL FDBSET(2,'READONLY',,,7)
C        DEFINE FILE 2(7,256,U,IREC)
C        IREC=1
C        DO 100 I=1,7
C        READ(2'IREC)(KA(I,J),J=1,20),(KB(I,J),J=1,20),
C     1   (KQ(I,J),J=1,20),(KZ(I,J),J=1,20),
C     1   (KM(I,J),J=1,20)
C 100    CONTINUE
C
C
         SUBROUTINE COMCAL(INT,X,KALPHA,KBETA,KQ,KZERO,KM,PATM,ALPHA,BETA,
      1   ALPHAT,BETAT,THETA,PHI,CTOT,CSTAT,CA,CB,CQ,CZERO,CDYN,
      2   CM,RMACH,U,V)
C        INT IS NO LONGER USER
C        X IS THE INPUT DATA ARRAY WITH PRESSURES IN IT
C        KALPHA, KBETA, KQ, KZERO, AND KM ARE THE PROBE CALIBRATION COE
C        PATM IS THE ATMOSPHERIC PRESSURE IN PSIA
C        ALPHA, BETA, ALPHAT, BETAT, THETA, AND PHI ARE THE ANGLES
C               RETURNED BY THE SUBROUTINE
C        CTOTAL, CSTATIC, CZERO, CQ, AND CDYN ARE THE PRESSURE COEFFICIENT
C               RETURNED BY THE SUBROUTINE
C        CA AND CB ARE THE ANGLE OF ATTACK COEFFICIENTS RETURNED
C        RMACH IS THE MACH NUMBER RETURNED BY THE SUBROUTINE
C        U AND V ARE THE RELATIVE VELOCITY VECTORS RETURNED
C
C
         REAL KALPHA,KBETA,KQ,KZERO,KM
         DIMENSION X(12),KALPHA(7,20),KBETA(7,20),KQ(7,20),KZERO(7,20),
      1   ICHAN(9),P(9),ICA(20),ICB(20),ICM(20),KM(7,20)
C        PRESETS TO DETERMINE WHICH PRESSURE IS WHICH INDEX INTO ARRAY X
         DATA ICHAN/2,3,4,5,6,7,10,8,9/
C        POWERS FOR CA IN THE POWER SERIES EXPANSION
         DATA ICA/0,1,0,0,2,0,0,1,0,1,3,0,0,2,2,1,0,1,0,1/
C        POWERS FOR CB IN THE POWER SERIES EXPANSION
         DATA ICB/0,0,1,0,0,2,0,1,1,0,0,3,0,1,0,2,2,0,1,1/
C        POWERS FOR CM IN THE POWER SERIES EXPANSION
         DATA ICM/0,0,0,1,0,0,2,0,1,1,0,0,3,0,1,0,1,2,2,1/
C        CONVERSION CONSTANT FOR RADIANS/DEGREES
         DATA RAD/57.296/
C        INTEGER FUNCTION TO DETERMINE SECTOR NUMBER
         ISEC(J)=MOD(J-1,6)+1
C        FUNCTION TO FIX RANGE OF ANGLES TO -90 TO 270
         RANGE(A)=AMOD(A+90.,360.)-90.
C        INITIALIZE THE HIGH PRESSURE VALUE TO A VERY NEGATIVE NUMBER
         PH=-1E20
C        PROCESS EACH PRESSURE
```

```
          DO 5 I=1,7
C         CONVERT THE PRESSURE TO ABSOLUTE - PSIA
          P(I)=X(ICHAN(I))+PATM
C         FIND THE MAXIMUM PRESSURE VALUE
 5        PH=AMAX1(PH,P(I))
C         CONVERT TOTAL PRESSURE TO ABSOLUTE
          P(8)=X(ICHAN(8))+PATM
C         CONVERT STATIC PRESSURE TO ABSOLUTE
          P(9)=X(ICHAN(9))+PATM
C         IF THE HIGH PRESSURE IS NOT FROM HOLE 7 SKIP THIS SECTION
          IF(PH.NE.P(7))GO TO 20
C         THE HIGH PRESSURE IS FROM SECTOR SEVEN - COMPUTE COEFFICIENTS
          ISECT=7
C         COMPUTE AVERAGE PRESSURE
          PAVG=(P(1)+P(2)+P(3)+P(4)+P(5)+P(6))/6.
C         COMPUTE THE VALUES TO DETERMINE CA AND CB
          CA1=(P(4)-P(1))/(PH-PAVG)
          CA2=(P(3)-P(6))/(PH-PAVG)
          CA3=(P(2)-P(5))/(PH-PAVG)
C         COMPUTE CA
          CA=CA1+(CA2-CA3)/2.
C         COMPUTE CB
          CB=.57735*(CA2+CA3)
C         COMPUTE CM
          CM=(P(7)-PAVG)/P(7)
C         SKIP THE NEXT SECTION WHICH IS FOR THE OUTER SECTORS
          GO TO 50
 20       DO 25 I=1,6
C         DETERMINE WHICH OUTER SECTOR HAS THE HIGH PRESSURE
          IF(P(I).EQ.PH)GO TO 30
 25       CONTINUE
C         SET THE SECTOR NUMBER
 30       ISECT=I
C         COMPUTE AVERAGE OF THE TWO HOLES ON EITHER SIDE OF HIGH PRES.
          PAVG=(P(ISEC(I+1))+P(ISEC(I+5)))/2.
C         COMPUTE CA
          CA=(P(I)-P(7))/(P(I)-PAVG)
C         COMPUTE CB
          CB=(P(ISEC(I+5))-P(ISEC(I+1)))/(P(I)-PAVG)
C         COMPUTE CM
          CM=(P(I)-PAVG)/P(I)
C         COMPUTE ALPHA, BETA, CQ, CZERO, AND MACH FROM CALIBRATION
 50       ALPHA=0.
          BETA=0.
          CQ=0.
          CZERO=0.
          RMACH=0.
C         USE THE POWER SERIES EXPANSION AND THE COEFFICIENTS TO
C         COMPUTE THE ACTUAL VALUES
          DO 60 I=1,20
          COEFF=CA**ICA(I)*CB**ICB(I)*CM**ICM(I)
          ALPHA=ALPHA+COEFF*KALPHA(ISECT,I)
          BETA=BETA+COEFF*KBETA(ISECT,I)
          CQ=CQ+COEFF*KQ(ISECT,I)
          CZERO=CZERO+COEFF*KZERO(ISECT,I)
          RMACH=RMACH+COEFF*KM(ISECT,I)
 60       CONTINUE
C         COMPUTE TOTAL PRESSURE COEFFICIENT
          CTOT=-CZERO*(PH-PAVG)/(P(8)-P(9))+(PH-P(8))/(P(8)-P(9))
C         COMPUTE STATIC PRESSURE COEFFICIENT
          CSTAT=-CZERO*(PH-PAVG)/(P(8)-P(9))-(PH-PAVG)/(CQ*
     1    (P(8)-P(9)))+(PH-P(9))/(P(8)-P(9))
C        COMPUTE DYNAMIC PRESSURE COEFFICIENT
         CDYN=CTOT-CSTAT
C        IF THIS IS AN OUTER SECTOR THEN SKIP THIS SECTION
         IF(PH.NE.P(7))GO TO 80
C        COMPUTE ALPHAT
         ALPHAT=ALPHA
C        COMPUTE BETAT
         BETAT=BETA
C        COMPUTE BETA
         BETA=ATAN(COS(ALPHA/RAD)*TAN(BETAT/RAD))*RAD
C        COMPUTE PHI
         IF(ALPHAT.NE.0.)PHI=AATAN(SIN(BETAT/RAD)*COS(ALPHAT/RAD),
     1    COS(BETAT/RAD)*SIN(ALPHAT/RAD))*RAD
         IF(ALPHAT.LT.0.)PHI=PHI+180.
         IF(ALPHAT.EQ.0..AND.BETAT.GE.0.)PHI=90.
         IF(ALPHAT.EQ.0..AND.BETAT.LT.0.)PHI=-90.
C        COMPUTE THETA
         THETA=ATAN(SQRT((TAN(ALPHAT/RAD))**2+(TAN(BETAT/RAD))**2))*RAD
C        SKIP THE SECTION FOR OUTER SECTORS
         GO TO 100
C        COMPUTE THETA
 80      THETA=ALPHA
C        COMPUTE PHI
         PHI=BETA
C        COMPUTE ALPHAT
         ALPHAT=AATAN(SIN(THETA/RAD)*COS(PHI/RAD),COS(THETA/RAD))*RAD
C        COMPUTE BETAT
         BETAT=AATAN(SIN(THETA/RAD)*SIN(PHI/RAD),COS(THETA/RAD))*RAD
C        COMPUTE ALPHA
         ALPHA=ALPHAT
C        COMPUTE BETA
         BETA=AATAN(COS(ALPHA/RAD)*SIN(THETA/RAD)*SIN(PHI/RAD),
     1    COS(THETA/RAD))*RAD
 100     CONTINUE
C        CHECK THE RANGE OF ANGLES FOR ALPHA, BETA, ALPHAT, BETAT, THETA,
         ALPHA=RANGE(ALPHA)
         BETA=RANGE(BETA)
         ALPHAT=RANGE(ALPHAT)
         BETAT=RANGE(BETAT)
         THETA=RANGE(THETA)
         PHI=RANGE(PHI)
C        COMPUTE THE ANGLE TANGENTS FOR USE IN VELOCITY VECTOR COMPUTATION
         TANAL=SIN(ALPHAT/57.296)/COS(ALPHAT/57.296)
         TANBE=SIN(BETAT/57.296)/COS(BETAT/57.296)
C        INSURE THAT CDYN DOES NOT EXCEED LIMITS
         IF (CDYN.LT.-.9999)CDYN=-.9999
C        COMPUTE THE LENGTH OF THE VELOCITY VECTOR
         UVR=SQRT((CDYN+1.)/(1.+TANAL*TANAL+TANBE*TANBE))
C        GET THE U COMPONENT OF THE VELOCITY VECTOR
         U=UVR*TANBE
C        GET THE V COMPONENT OF THE VELOCITY VECTOR
         V=UVR*TANAL
         RETURN
         END
         FUNCTION AATAN(T,B)
         ATAN=ATAN2(T,B)
         IF(T.GE.0..AND.B.GE.0.)AATAN=ATAN
         IF(T.GE.0..AND.B.LT.0.)AATAN=3.14159+ATAN
         IF(T.LT.0..AND.B.GE.0.)AATAN=2.*3.14159+ATAN
         IF(T.LT.0..AND.B.LT.0.)AATAN=3.14159+ATAN
```

```
      RETURN
      END
      FUNCTION TAN(A)
      TAN=SIN(A)/COS(A)
      RETURN
      END
```

```
01234567890123456789   ** RSX-11M V3.2 **   19-SEP-81   14:26:38   DF0:[152,13]FILEIO.FTN;7   01234567890123456789
01234567890123456789   ** RSX-11M V3.2 **   18-SEP-81   14:26:38   DF0:[152,13]FILEIO.FTN;7   01234567890123456789
01234567890123456789   ** RSX-11M V3.2 **   18-SEP-81   14:26:38   DF0:[152,13]FILEIO.FTN;7   01234567890123456789


FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    IIIIII      000000
FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    IIIIII      000000
FF              II      LL          EE              II      00    00
FF              II      LL          EE              II      00    00
FF              II      LL          EE              II      00    00
FF              II      LL          EE              II      00    00
FFFFFFFF        II      LL          EEEEEEEE        II      00    00
FFFFFFFF        II      LL          EEEEEEEE        II      00    00
FF              II      LL          EE              II      00    00
FF              II      LL          EE              II      00    00
FF              II      LL          EE              II      00    00
FF              II      LL          EE              II      00    00      ....
FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE    IIIIII      000000        ....
FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE    IIIIII      000000        ....
                                                                            ....


FFFFFFFFFF  TTTTTTTTTT  NN      NN    ;;;;    77777777
FFFFFFFFFF  TTTTTTTTTT  NN      NN    ;;;;    77777777
FF              TT      NN      NN    ;;;;          77
FF              TT      NN      NN    ;;;;          77
FF              TT      NNNN    NN                  77
FF              TT      NNNN    NN                  77
FFFFFFFF        TT      NN  NN  NN    ;;;;        77
FFFFFFFF        TT      NN  NN  NN    ;;;;        77
FF              TT      NN    NNNN    ;;;;      77
FF              TT      NN    NNNN    ;;;;      77
FF              TT      NN      NN      ;;    77
FF              TT      NN      NN      ;;    77
FF              TT      NN      NN    ;;    77
FF              TT      NN      NN    ;;    77


01234567890123456789   ** RSX-11M V3.2 **   18-SEP-81   14:26:38   DF0:[152,13]FILEIO.FTN;7   01234567890123456789
01234567890123456789   ** RSX-11M V3.2 **   18-SEP-81   14:26:38   DF0:[152,13]FILEIO.FTN;7   01234567890123456789
01234567890123456789   ** RSX-11M V3.2 **   19-SEP-81   14:26:38   DF0:[152,13]FILEIO.FTN;7   01234567890123456789
```

```
C################################################################################
C
C
C       ROUTINE TO GET A GRID OF DATA FROM A FILE IN THE STANDARD
C       DFAN DATA FILE FORMAT.
C
C
C##############################################################################
        SUBROUTINE GETGRD(LUN,ND,IDLIST,MAXSIZ,LSCR,RSCR,STITLE,LTITLE,
     1   DLABEL,DAT,NP,NL,NG,NLAT,EOF)
C
C
C       LUN                     LOGICAL UNIT NUMBER OF DATA FILE TO BE WRITTEN
C       ND                      NUMBER OF DIMENSIONS IN EACH DATA POINT
C       NP                      NUMBER OF POINTS IN A LINE
C       NL                      NUMBER OF LINES IN A GRID
C       NG                      GRID NUMBER
C       NLAT                    LATTICE NUMBER
C       DLABEL                  STORAGE SPACE FOR DIMENSION LABELS TO BE SAVED
C       DAT                     DATA POINTS ARRAY (ND,NL*NL)
C       STITLE                  STORAGE SPACE FOR 20 CHARACTER SHORT TITLE
C       LTITLE                  STORAGE SPACE FOR 60 CHARACTER LONG TITLE
C       IDLIST                  ARRAY OF INDICES OF DATA TO BE SAVED
C       MAXSIZ                  DIMENSION OF SECOND INDEX OF DAT ARRAY
C       LSCR                    LOGICAL*1 SCRATCH ARRAY (10*MAXSIZ)
C       RSCR                    REAL SCRATCH ARRAY (MAXSIZ)
C       EOF                     LOGICAL END-OF-FILE FLAG - TRUE WHEN EOF WAS READ
C
        LOGICAL*1 STITLE(20),LTITLE(60),DLABEL(10,ND),LSCR(1)
        LOGICAL EOF
        DIMENSION DAT(ND,MAXSIZ),RSCR(1),IDLIST(ND)
 11     FORMAT(5I5)
C       READ LINES FROM FILE UNTIL THE GRID NUMBER CHANGES
        NP=0
        NL=0
        NTOT=0
 10     CALL GETLIN(LUN,ND,IDLIST,MAXSIZ,LSCR,RSCR,STITLE,LTITLE,DLABEL,
     1   DAT(1,NTOT+1),NP1,NL1,NG,NLAT,EOF)
        NP=NP+NP1
        NL=NL+NL1
        NTOT=NTOT+NP1*NL1
        READ(LUN,11,END=999)MND,NP2,NL2,NG2,NLAT2
        BACKSPACE LUN
        IF(NG2.EQ.NG)GO TO 10
        RETURN
 999    CONTINUE
        RETURN
        END
C################################################################################:
C
C
C       ROUTINE TO GET A LINE OF DATA FROM A FILE IN THE STANDARD
C       DFAN DATA FILE FORMAT.
C
C
C#############################################################################
        SUBROUTINE GETLIN(LUN,ND,IDLIST,MAXSIZ,LSCR,RSCR,STITLE,LTITLE,
     1   DLABEL,DAT,NP,NL,NG,NLAT,EOF)
C
C
```

```
C       LUN                 LOGICAL UNIT NUMBER OF FILE TO BE READ(INPUT)
C       ND                  NUMBER OF VARIABLES TO GET FROM DATA POINT(INPUT)
C       IDLIST              ARRAY OF INDICES OF VARIABLES TO BE RETURNED
C       MAXSIZ              MAXIMUM SIZE OF SECOND DIMENSION OF DAT ARRAY
C       LSCR                LOGICAL SCRATCH ARRAY (10 X NUMBER OF DIMENSIONS
C                           IN EACH DATA POINT)
C       RSCR                REAL SCRATCH ARRAY (NUMBER OF DIMENSIONS IN EACH
C                           DATA POINT LONG)
C       STITLE              SHORT TITLE (20 CHARACTERS) (OUTPUT)
C       LTITLE              LONG TITLE (60 CHARACTERS) (OUTPUT)
C       DLABEL              DIMENSION LABELS ARRAY (10,ND) (OUTPUT)
C       DAT                 ARRAY OF DATA RETURNED (ND,MAXSIZ) (OUTPUT)
C       NP                  NUMBER OF POINTS IN THE LINE(OUTPUT)
C       NL                  NUMBER OF LINES IN GRID (OUTPUT)
C       NG                  GRID NUMBER (OUTPUT)
C       NLAT                LATTICE NUMBER (OUTPUT)
C       EOF                 LOGICAL FLAG - TRUE IF END OF FILE ENCOUNTERED
C
        LOGICAL*1 DLABEL(10,ND),STITLE(20),LTITLE(60),LSCR
        LOGICAL EOF
        DIMENSION DAT(ND,MAXSIZ),RSCR(1),IDLIST(ND)
 11     FORMAT(5I5)
 12     FORMAT(10E12.5)
        EOF=.FALSE.
C       GET THE NUMBER OF DIMENSIONS PER POINT, NUMBER OF POINTS
C       PER LINE, NUMBER OF LINES, GRID NUMBER, AND LATTICE NUMBER
        READ(LUN,11,END=999)MND,NP,NL,NG,NLAT
C       GET THE HEADER FOR THE LINE
        CALL GETHDR(LUN,MND,NB,IDLIST,LSCR,DLABEL,STITLE,LTITLE,EOF)
        IF(EOF)RETURN
C       COMPUTE NUMBER OF POINTS IN LINE/GRID
        NL=MAX0(NL,1)
        NTOT=NL*NP
C       GET THE DATA POINTS FOR THE LINE/GRID
        DO 100 I=1,NTOT
        READ(LUN,12,END=999)(RSCR(J),J=1,MND)
C       IF THE DAT ARRAY IS FULL, DO NOT SAVE THIS POINT
        IF(I.GT.MAXSIZ)GO TO 100
        DO 50 J=1,ND
        DAT(J,I)=RSCR(IDLIST(J))
 50     CONTINUE
 100    CONTINUE
        RETURN
 999    CONTINUE
        EOF=.TRUE.
        RETURN
        END
C************************************************************************
C
C
C       ROUTINE TO GET THE HEADER RECORDS FROM A FILE IN THE STANDARD
C       DFAN DATA FILE FORMAT.
C
C
C**********************************************************************
        SUBROUTINE GETHDR(LUN,MND,ND,IDLIST,LSCR,DLABEL,STITLE,LTITLE,EOF
C
C
C       LUN                 LOGICAL UNIT NUMBER OF DATA FILE TO BE READ
C       MND                 NUMBER OF DIMENSIONS IN A DATA POINT
C       ND                  NUMBER OF DIMENSIONS TO SAVE FROM EACH DATA POINT
```

```
C       IDLIST          ARRAY OF INDICES OF VARIABLES TO BE SAVED
C       LSCR            LOGICAL*1 SCRATCH ARRAY FOR USE IN READ
C       DLABEL          STORAGE SPACE FOR DIMENSION LABELS TO BE SAVED
C       STITLE          STORAGE SPACE FOR 20 CHARACTER SHORT TITLE
C       LTITLE          STORAGE SPACE FOR 60 CHARACTER LONG TITLE
C       EOF             END-OF-FILE FLAG - SET TO TRUE IF EOF ENCOU
C
        DIMENSION IDLIST(ND)
        LOGICAL*1 STITLE(20),LTITLE(60),LSCR(10,MND),DLABEL(10,ND)
        LOGICAL EOF
 11     FORMAT(20A1,60A1)
 12     FORMAT(10A1)
C       READ RECORD 2 - GET THE TITLES
        READ(LUN,11,END=999)STITLE,LTITLE
C       READ RECORD 3 - GET ALL DIMENSION LABELS
        READ(LUN,12,END=999)((LSCR(I,J),I=1,10),J=1,MND)
C       SAVE THE DESIRED DIMENSION LABELS IN DLABEL
        DO 100 J=1,ND
        DO 50 I=1,10
        DLABEL(I,J)=LSCR(I,IDLIST(J))
 50     CONTINUE
 100    CONTINUE
        RETURN
 999    CONTINUE
        EOF=.TRUE.
        RETURN
        END
C*******************************************************************
C
C
C       ROUTINE TO PUT THE HEADER RECORDS ON A FILE IN THE STANDARD
C       DFAN DATA FILE FORMAT.
C
C
C*******************************************************************
        SUBROUTINE PUTHDR(LUN,ND,NP,NL,NG,NLAT,STITLE,LTITLE,DLABEL)
C
C
C       LUN             LOGICAL UNIT NUMBER FOR OUTPUT DATA
C       ND              NUMBER OF DIMENSIONS IN A DATA POINT
C       NP              NUMBER OF POINTS IN A LINE
C       NL              NUMBER OF LINES IN A GRID
C       NG              GRID NUMBER
C       NLAT            LATTICE NUMBER
C       STITLE          20 CHARACTER SHORT TITLE
C       LTITLE          60 CHARACTER LONG TITLE
C       DLABEL          10 CHARACTER DIMENSION LABELS
C
        LOGICAL*1 STITLE(20),LTITLE(60),DLABEL(10,ND)
 11     FORMAT(5I5)
 12     FORMAT(20A1,60A1)
 13     FORMAT(10A1)
C       OUTPUT RECORD #1
        WRITE(LUN,11)ND,NP,NL,NG,NLAT
C       OUTPUT RECORD #2
        WRITE(LUN,12)STITLE,LTITLE
C       OUTPUT RECORD #3
        WRITE(LUN,13)((DLABEL(I,J),I=1,10),J=1,ND)
        RETURN
        END
```

```
01234567890123456789     ** RSX-11M V3.2 **     18-SEP-81     14:27:58     DF0:[152,13]GRAFS.FTN;250     01234567890123456789
01234567890123456789     ** RSX-11M V3.2 **     18-SEP-81     14:27:58     DF0:[152,13]GRAFS.FTN;250     01234567890123456789
01234567890123456789     ** RSX-11M V3.2 **     18-SEP-81     14:27:58     DF0:[152,13]GRAFS.FTN;250     01234567890123456789



  GGGGGGGG   RRRRRRRR       AAAAAA     FFFFFFFFFF     SSSSSSSS
  GGGGGGGG   RRRRRRRR       AAAAAA     FFFFFFFFFF     SSSSSSSS
GG           RR      RR   AA      AA   FF           SS
GG           RR      RR   AA      AA   FF           SS
GG           RR      RR   AA      AA   FF           SS
GG           RR      RR   AA      AA   FF           SS
GG           RRRRRRRR     AA      AA   FFFFFFFF       SSSSSS
GG           RRRRRRRR     AA      AA   FFFFFFFF       SSSSSS
GG  GGGGGG   RR  RR       AAAAAAAAAA   FF                   SS
GG  GGGGGG   RR  RR       AAAAAAAAAA   FF                   SS
GG      GG   RR    RR     AA      AA   FF                   SS        ....
GG      GG   RR    RR     AA      AA   FF                   SS        ....
  GGGGGG     RR      RR   AA      AA   FF           SSSSSSSS          ....
  GGGGGG     RR      RR   AA      AA   FF           SSSSSSSS          ....



FFFFFFFFFF   TTTTTTTTTT   NN      NN       ;;;;       222222     5555555555     000000
FFFFFFFFFF   TTTTTTTTTT   NN      NN       ;;;;       222222     5555555555     000000
FF               TT       NN      NN       ;;;;     22      22   55           00     00
FF               TT       NN      NN       ;;;;     22      22   55           00     00
FF               TT       NNNN    NN                        22   555555       00   0000
FF               TT       NNNN    NN                        22   555555       00   0000
FFFFFFFF         TT       NN  NN  NN       ;;;;           22           55     00  00 00
FFFFFFFF         TT       NN  NN  NN       ;;;;           22           55     00  00 00
FF               TT       NN    NNNN       ;;;;         22               55   0000   00
FF               TT       NN    NNNN       ;;;;         22               55   0000   00
FF               TT       NN      NN         ;;       22         55      55   00     00
FF               TT       NN      NN         ;;       22         55      55   00     00
FF               TT       NN      NN       ;;       2222222222     555555       000000
FF               TT       NN      NN       ;;       2222222222     555555       000000



01234567890123456789     ** RSX-11M V3.2 **     18-SEP-81     14:27:58     DF0:[152,13]GRAFS.FTN;250     01234567890123456789
01234567890123456789     ** RSX-11M V3.2 **     18-SEP-81     14:27:58     DF0:[152,13]GRAFS.FTN;250     01234567890123456789
01234567890123456789     ** RSX 11M V3.2 **     18-SEP-81     14:27:58     DF0:[152,13]GRAFS.FTN;250     01234567890123456789
```

```
C
C
C       PLOTTING PROGRAM FOR CONTOUR, AXONOMETRIC, AND VECTOR
C       FIELD PLOTS - THIS PROGRAM ACCESSES DATA IN DATA FILE
C       'DATAIN.DAT' WHICH IS IN THE STANDARD DFAN DATA FILE
C       FORMAT.  THE PROGRAM IS AN EXTENSION OF 'FIELDPLT' WHICH
C       WAS DEVELOPED AND WRITTEN BY CAPT GLYNN SISSON.  THE
C       PROGRAM HAS BEEN MODIFIED AS FOLLOWS:
C
C       MODIFICATION RECORD:
C
C       INITIAL MODS AND COMMENTS        29 JULY 81       CAPT BOLICK
C       MORE COMMENTS ADDED     18 SEPT 81       CAPT BOLICK
C
C
C******************************************************************
C
C
C       DATA DECLARATION AREA
C
C
C******************************************************************
        DIMENSION X1(10),Y1(10),Z1(10),V1(10),X2(10),Y2(10),
     1          Z2(10),V2(10)
        DIMENSION LINE(20),X3(50),Y3(50),PARM(31)
        DIMENSION F(4,2),DAT(5,100),IDLIST(5)
        LOGICAL*1 LSCR(10,17),STITLE(20),LTITLE(60),NADE(10),DLABEL(10,5
        LOGICAL EOF
        LOGICAL*1 TITLE(60)
        EQUIVALENCE (LSCR(1,1),X3(1))
        EQUIVALENCE (N1,IDLIST(1)),(N2,IDLIST(2)),(N3,IDLIST(3)),
     1   (N4,IDLIST(4)),(N5,IDLIST(5))
C******************************************************************
C
C
C       DATA PRESETS AREA
C
C
C*******************************************************************
        DATA YES/'Y'/
C       DISCRIPTOR FOR DIFFERENT LINE TYPES TO BE DRAWN
        DATA LINE/77,12,32,332,512,5312,72,732,34,334,514,
     *    5314,74,734,3212,5212,7212,3414,5414,7414/
        DATA LUN/2/,MAXSIZ/100/,IBAUD/960/,ND/5/
C********************************************************************
C
C
C       MAIN PROGRAM CODE STARTS HERE
C
C
C************************************************************
C
C
C       GET MENU AND INPUT DATA FILES ASSIGNED
C       SET UP TERMINAL LOGICAL UNITS ALSO
C
C
        CALL ASSIGN(3,'MENU.DAT',8)
        CALL FDBSET(3,'READONLY')
C       READ THE INPUT DATA FROM THE MENU DATA FILE
```

```
        READ(3)TITLE,OPT,AP,F,XMIN,XMAX,XINC,YMIN,YMAX,YINC,
     1          ZMIN,ZMAX,ZINC,IDLIST
C       SAVE THE SCALE FACTOR FOR VECTOR PLOTS
        SCALEF=F(3,2)
C       CLEAR THE FIELD IF THE OPTION IS VECTOR FIELD PLOT
        IF (OPT.GT.1.)F(3,2)=0.
C       THE OUTPUT CRT FOR THE GRAPH WILL BE 'TT7:'
        CALL ASSIGN(5,'TT7:')
C       IF A TK 4662 HARDCOPY WAS REQUESTED CHANGE UNIT 5 TO 'TT1:'
        IF(AP.EQ.YES)CALL ASSIGN(5,'TT1:')
C       INITIALIZE THE GRAPHICS PACKAGE - CLEAR THE SCREEN
        CALL INITT(IBAUD)
C       GET THE INPUT DATA FILE ASSIGNED
        CALL ASSIGN(2,'DATAIN.DAT',10)
        CALL FDBSET(2,'OLD','SHARE')
C       FILL THE F ARRAY BASED ON THE MIN/MAX/ANGLE DATA
        CALL FFILL(F,XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX,SXMIN,SXMAX,
     1          SYMIN,SYMAX)
C       IF THIS IS NOT A CONTOUR PLOT THEN SKIP TO THE NEXT SECTION
        IF(OPT.NE.0.)GO TO 200
C       SET UP THE PLOTTING MINMUM AND MAXIMUM VALUES AND DEFINE
C       THE PLOTTING WINDOW FOR THE CONTOUR PLOT.
        RANGE=SXMAX-SXMIN
        XMI=SXMIN-.05*RANGE
        XMA=SXMAX+.45*RANGE
        YMI=SYMIN-.05*RANGE
        YMA=SYMAX+.1*RANGE
        CALL DWINDO(XMI,XMA,YMI,YMA)
        CALL TWINDO(0,1020,0,780)
        ZLOW=ZMIN
C**************************************************************
C
C
C       CONTOUR PLOTS ARE DONE HERE
C
C
C****************************************************************
 100    CONTINUE
C       GET THE NEXT GRID/PLANE OF DATA
        CALL GETGRD(LUN,ND,IDLIST,MAXSIZ,LSCR,X3,STITLE,LTITLE,DLABEL,
     1   DAT,NP,NL,NG,NLAT,EOF)
C       IF NO MORE DATA THEN PROCEED TO FINISH UP
        IF(EOF)GO TO 130
        ICNT=0
        Z3=ZLOW
110     CONTINUE
        ICNT=ICNT+1
        K1=1
        K2=NP+1
        K=K2
        DO 120 J=2,NL
C       MOVE A LINE OF DATA AT A TIME INTO THE PLOTTING ARRAYS
        CALL FILLIT(NP,K1,K2,X1,Y1,Z1,V1,X2,Y2,Z2,V2,DAT)
C       GO DO THIS CONTOUR LINE
        CALL CONTOR(NP,X1,Y1,Z1,V1,X2,Y2,Z2,V2,X3,Y3,Z3,LINE(ICNT))
        K1=K1+NP
        K2=K2+NP
120     CONTINUE
C       SET UP FOR NEXT CONTOUR LINE
        Z3=Z3+ZINC
C       IF NOT ALL CONTOURS ARE PLOTTED THEN PLOT THE NEXT ONE
```

```
      IF(Z3.LE.ZMAX)GO TO 110
C     AFTER ALL CONTOURS ON THIS PLANE ARE PLOTTED - GO SEE IF
C     THERE IS ANOTHER PLANE OF DATA TO PLOT
      GO TO 100
 130  CONTINUE
C     COMPUTE LOCATIONS FOR THE LEGEND ON THE GRAPH
      X=SXMAX+.1*(SXMAX-SXMIN)
      Y=SYMAX-.05*(SYMAX-SYMIN)
      XI=.25*(SXMAX-SXMIN)
      YI=.04*(SYMAX-SYMIN)
C     GO PUT LEGEND ON THE GRAPH
      CALL LEGEND(ICNT,LINE,ZLOW,ZINC,X,Y,XI,YI)
C     COMPUTE LOCATION FOR NAME OF VARIABLE PLOTTED
      X=.5*(SXMAX-SXMIN)+SXMIN
      Y=SYMAX+.01*(SXMAX-SXMIN)
      CALL MOVEA(X,Y)
      CALL ANMODE
C     OUTPUT VARIABLE NAME THAT WAS PLOTTED
      CALL AANSTR(10,DLABEL(1,3))
C     GO TO FINISH UP GRAPH
      GO TO 500
C*******************************************************
C
C
C           AXONOMETRIC (3-D) PLOTS ARE DONE HERE
C
C
C********************************************************
200   CONTINUE
C     IF AN AXONOMETRIC PLOT WAS NOT REQUESTED THEN GO TO CROSS FLOW
C     PLOT SECTION
      IF(OPT.NE.1.)GO TO 300
C     COMPUTE RANGE AND LIMITS FOR THE AXONOMETRIC PROJECTION AND
C     THEN SET THE PROPER LIMITS IN THE WINDOW DEFINITIONS SO THAT
C     ALL OF THE GRAPH WILL BE ON THE SCREEN
      RANGE=AMAX1(SXMAX,SYMAX)-AMIN1(SXMIN,SYMIN)
      XMI=SXMIN-.05*RANGE
      XMA=SXMIN+1.2*RANGE
      YMI=SYMIN-.05*RANGE
      YMA=SYMIN+1.1*RANGE
      CALL DWINDO(XMI,XMA,YMI,YMA)
      CALL TWINDO(125,975,0,780)
 210  CONTINUE
C     GET THE NEXT GRID/PLANE OF DATA
      CALL GETGRD(LUN,ND,IDLIST,MAXSIZ,LSCR,X3,STITLE,LTITLE,DLABEL,
     1 DAT,NP,NL,NG,NLAT,EOF)
C     IF THERE IS NO MORE DATA THEN GO TO FINISH UP THIS PLOT
      IF(EOF)GO TO 250
C     COMPUTE TOTAL NUMBER OF DATA POINTS IN THE GRID
      NTOT=NP*NL
      K1=1
      K2=NP+1
      DO 220 J=2,NL
C     MOVE THE DATA INTO THE PLOTTING ARRAYS ONE LINE AT A TIME
      CALL FILLIT(NP,K1,K2,X1,Y1,Z1,V1,X2,Y2,Z2,V2,DAT)
C     IF THIS IS THE FIRST RAIL FOR THIS PLANE THEN CALL D3MESH
C     TO PROPERLY SET UP TO DRAW THE MESH
      IF(J.EQ.2)CALL D3MESH(F,NP,X1,Y1,Z1,V1,X1,Y1,Z1,V1)
C     DRAW THE NEXT SET OF RAILS FOR THIS PLANE
      CALL D3MESH(F,NP,X1,Y1,Z1,V1,X2,Y2,Z2,V2)
      K1=K1+NP
      K2=K2+NP
220   CONTINUE
C     GO SEE IF THERE IS ANY MORE DATA TO BE PLOTTED
      GO TO 210
 250  CONTINUE
C     COMPUTE LOCATION TO OUTPUT NAME OF VARIABLE THAT WAS PLOTTED
      X=.5*(SXMAX-SXMIN)+SXMIN
      Y=SYMAX+.01*(SXMAX-SXMIN)
      CALL MOVEA(X,Y)
      CALL ANMODE
C     OUTPUT VARIABLE NAME
      CALL AANSTR(10,DLABEL(1,3))
      GO TO 500
C*****************************************************************
C
C
C     VECTOR FIELD PLOTS ARE DONE HERE
C
C
C******************************************************************
 300  CONTINUE
C     COMPUTE RANGE AND LIMITS FOR PLOTTING WINDOWS FOR THE CROSS
C     FLOW VECTOR PLOTS.
      RANGE=SXMAX-SXMIN
      XMI=SXMIN-.05*RANGE
      XMA=SXMAX+.2*RANGE
      YMI=SYMIN-.05*RANGE
      YMA=SYMAX+.1*RANGE
      CALL DWINDO(XMI,XMA,YMI,YMA)
      CALL TWINDO(125,975,0,780)
 310  CONTINUE
C     GET THE NEXT GRID/PLANE OF DATA
      CALL GETGRD(LUN,ND,IDLIST,MAXSIZ,LSCR,X3,STITLE,LTITLE,DLABEL,
     1 DAT,NP,NL,NG,NLAT,EOF)
C     IF NO MORE DATA THEN GO TO FINISH UP THE GRAPH
      IF(EOF)GO TO 500
      NTOT=NP*NL
C     CALL THE SUBROUTINE TO PLOT EACH OF THE VECTORS
      CALL VFIELD(NTOT,DAT,SCALEF)
C     GO SEE IF THERE IS MORE DATA TO BE PLOTTED
      GO TO 310
500   CONTINUE
C     OUTPUT GRID AROUND THE PLOT SPACE
      CALL GRID(F:XMIN,XMAX,XINC,YMIN,YMAX,YINC)
      CALL MOVABS(100,760)
      TITLE(46)=32
      TITLE(47)=32
C     OUTPUT THE TITLE OF THE GRAPH
      CALL AANSTR(60,TITLE)
C     CLOSE OUT THE PLOTTING PACKAGE
      CALL FINITT(0,0)
      CALL EXIT
      END
```

APPENDIX B

This appendix contains samples of the user interactions required for both the data acquisition and plotting programs.

```
>HELLO
ACCOUNT OR NAME: GRIFFIN
PASSWORD:

        RSX-11M BL26   MULTI-USER SYSTEM

GOOD MORNING
06-OCT-81 09:31 LOGGED ON TERMINAL TT0:

WELCOME TO RSX-11M V3.2 REV D FOR DFAN

>@[152,13]CMPSHP
>;*****************************************************************
>;
>;
>;       COMPRESSIBLE SEVEN HOLE PROBE FLOW FIELD SURVEY
>;       DATA ACQUISITION SYSTEM COMMAND FILE
>;
>;
>;*****************************************************************
>PIP TARE.DAT;*/DE/NM
>PIP MENU.DAT;*/DE/NM
>PIP DATAIN.DAT;*/DE/NM
>PIP DATAOUT.DAT;*/DE/NM
>* DO YOU WANT TO ADD DATA TO AN EXISTING FILE? [Y/N]:Y
>* ENTER NAME OF CURRENT DATA FILE [S]: XWM4.DAT
>PIP DATAIN.DAT/NV=XWM4.DAT
>PIP TARE.DAT/NV=[152,13]NULL.DAT
>RUN [152,13]CMPSHP
THE PRESENT DATA TITLE IS:
 FWING-BODY, 11 DEG, V=100 FPS (XWM4)
DO YOU WISH TO CHANGE IT?
Y
ENTER DATA TITLE (40 CHARS MAX)
TEST RUN OF FLOW FIELD SURVEY PROGRAM
START TUNNEL AND ENTER RETURN WHEN READY

ENTER ATMOSPHERIC PRESSURE (IN-HG) .
29.40
COORDINATE PAIRS ARE (VERT,HORIZ)
ENTER LOCATION OF LOWER LEFT AND ENTER RETURN
-3,6
ENTER LOCATION OF UPPER LEFT AND ENTER RETURN
6,6
ENTER LOCATION OF LOWER RIGHT AND ENTER RETURN
-3,8
ENTER LOCATION OF UPPER RIGHT AND ENTER RETURN
7,8
LOWER LEFT:     -3.000      6.000
UPPER LEFT:      6.000      6.000
LOWER RIGHT:     -3.000      8.000
UPPER RIGHT:      7.000      8.000
ARE THESE LOCATIONS CORRECT?
Y
FIRST DATA POINT: VERTICAL=     -3.000 HORIZONTAL=      6.000
TT0  --  STOP
>PIP DATAIN.DAT/NV=DATAOUT.DAT
>PIP DATAOUT.DAT;*/DE/NM
>PIP DATAIN.DAT/PU
>* DO YOU WANT TO USE AN EXISTING MENU DATA FILE? [Y/N]:
 PIP MENU.DAT/NV=[132,132]NULL.DAT
 RUN [132,132]MENU

ENTER PLOT OPTION: 0=CONTOURS, 1=ISOMETRIC, 2=CROSS P...
0


   DATA XMIN =        1.000
DATA XMAX =        6.000
DATA YMIN =       -3.000
DATA YMAX =        7.000
DATA ZMIN =       -1.500
 DATA ZMAX =        1.404
1.OPTION = 0.
2.TITLE = TEST RUN OF FLOW FIELD SURVEY PROGRAM    07:32:12 26-OCT-81
3.X AXIS ANGLE =        0.000
4.Y AXIS ANGLE =       90.000
5.Z AXIS ANGLE =        0.000
6.X MIN =        0.000
7.X MAX =       10.000
8.Y MIN =       -4.000
9.Y MAX =        6.000
10. Z VARIABLE = 10. CTOTAL
11.ZMIN =    -1.50000
12.ZMAX =     1.40400
13.ZINC =     0.50000
ENTER NUMBER OF PARAMETER TO CHANGE AND NEW VALUE (IF APPROPRIATE)

DO YOU WANT A HARDCOPY PLOT?N
> [illegible]
        TT0  --  STOP
 PIP MENU.DAT/PU
 RUN [132,132]GRAPHS
TT0  --  STOP
 * DO YOU WANT ANOTHER GRAPH? [Y/N]:
 * DO YOU WANT TO TAKE MORE DATA? [Y/N]:
 * DO YOU WANT TO SAVE THE MENU DATA FILE? [Y/N]:
 PIP MENU.DAT;*/DE/NM
 * ENTER NAME FOR OUTPUT DATA FILE [S]:
 PIP SAVNAM.DAT/NV=DATAIN.DAT
 PIP DATAIN.DAT;*/DE/NM
 PIP TARE.DAT;*/DE/NM
 @ <EOF>
```

AD-A115 971 AIR FORCE ACADEMY CO F/G 9/2

SEVEN-HOLE PROBE DATA ACQUISITION SYSTEM.(U)

NOV 81 A GERNER, G SISSON

UNCLASSIFIED USAFA-TN-81-8 NL

2 OF 2

AD A

971

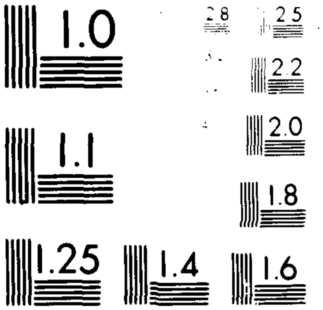
END

DATE

FILMED

08-82

DTIC

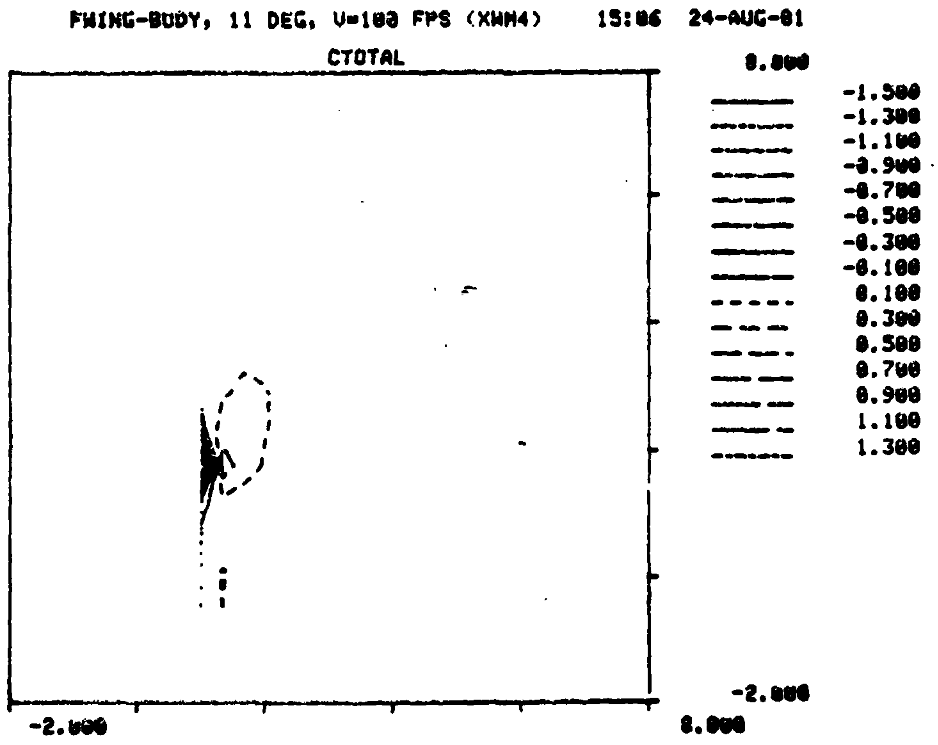


MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

```
15-SEP-81 08:43          FROM TT10:       TO TT0:


>HEL GRIFFIN/AERO

         RSX-11M BL26   MULTI-USER SYSTEM

GOOD MORNING
15-SEP-81 08:50 LOGGED ON TERMINAL TT0:

WELCOME TO RSX-11M V3.2 REV D FOR DFAN


@[152,13]PLOT
>;**************************************************************
>;
>;
>;      FLOW FIELD SURVEY PLOTING COMMAND FILE
>;
>;
>;**************************************************************
>* ENTER NAME OF DATA FILE TO BE PLOTTED [S]: XWM4.DAT
>PIP DATAIN.DAT/NV=XWM4.DAT
>* DO YOU WANT TO USE AN EXISTING MENU DATA FILE? [Y/N]:N
>PIP MENU.DAT/NV=[152,13]NULL.DAT
>RUN [152,13]MENU


 ENTER PLOT OPTION: 0=CONTOUR, 1=AXONOMETRIC, 2=CROSS FLOW
 0

  DATA XMIN -       1.000
DATA XMAX =      6.000
DATA YMIN =     -3.000
DATA YMAX =      7.000
DATA ZMIN =     -1.913
 DATA ZMAX =      1.405
1.OPTION = 0.
2.TITLE = FWING-BODY, 11 DEG, V=100 FPS (XWM4)    15:06:524-AUG-81
3.X AXIS ANGLE =       0.000
4.Y AXIS ANGLE =      90.000
5.Z AXIS ANGLE =       0.000
6.X MIN =        0.000
7.X MAX =       10.000
8.Y MIN =       -4.000
9.Y MAX =        6.000
10. Z VARIABLE = 10. CTOTAL
11.ZMIN =    -1.50000
12.ZMAX =     1.40450
13.ZINC =     0.50000
ENTER NUMBER OF PARAMETER TO CHANGE AND NEW VALUE (IF APPROPRIATE)
```

```
    .OPTION = 0.
 2.TITLE = FWING-BODY, 11 DEG, V=100 FPS (XWM4)     15:06:524-AUG-81
.3.X AXIS ANGLE =        0.000
 4.Y AXIS ANGLE =       90.000
 5.Z AXIS ANGLE =        0.000
 6.X MIN =      -2.000
 7.X MAX =       8.000
 8.Y MIN =      -2.000
 9.Y MAX =       8.000
 10. Z VARIABLE = 10.
 11.ZMIN =    -1.50000
 12.ZMAX =     1.40450
 13.ZINC =     0.20000
 ENTER NUMBER OF PARAMETER TO CHANGE AND NEW VALUE (IF APPROPRIATE)
 ■
 DO YOU WANT A HARDCOPY PLOT? N
 7 @$d @
        TTO  --  STOP
 >PIP MENU.DAT/PU
 >RUN [152,13]GRAFS
```

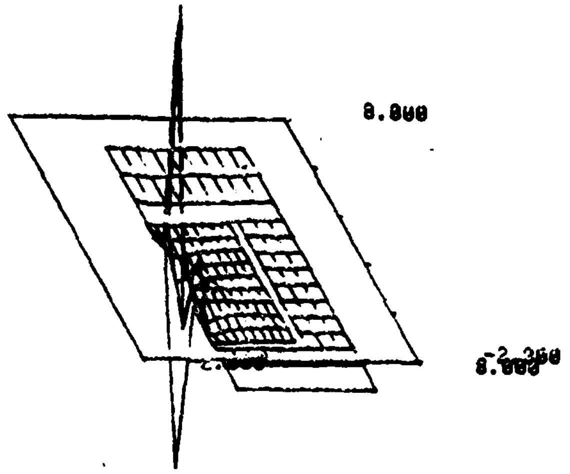


```
  TTO  --  STOP
 >* DO YOU WANT ANOTHER GRAPH? [Y/N]:Y
 >* DO YOU WANT TO USE A DIFFERENT MENU FILE? [Y/N]:N
 >RUN [152,13]MENU



    DO YOU WANT TO SEE THE DATA MINIMUM/MAXIMUM VALUES? [Y/N]
 ■
```

```
   .OPTION = 0.
2.TITLE = FWING-BODY, 11 DEG, V=100 FPS (XWM4)    15:06:524-AUG-81
3.X AXIS ANGLE =       0.000
4.Y AXIS ANGLE =      90.000
5.Z AXIS ANGLE =       0.000
6.X MIN =      -2.000
7.X MAX =       8.000
8.Y MIN =      -2.000
9.Y MAX =       8.000
10. Z VARIABLE = 10.
11.ZMIN =   -1.50000
12.ZMAX =    1.40450
13.ZINC =    0.20000
ENTER NUMBER OF PARAMETER TO CHANGE AND NEW VALUE (IF APPROPRIATE)
1,8



   .OPTION = 1.
2.TITLE = FWING-BODY, 11 DEG, V=100 FPS (XWM4)    15:06:524-AUG-81
3.X AXIS ANGLE =       0.000
4.Y AXIS ANGLE =     120.000
5.Z AXIS ANGLE =      90.000
6.X MIN =      -2.000
7.X MAX =       8.000
8.Y MIN =      -2.000
9.Y MAX =       8.000
10. Z VARIABLE = 10. CTOTAL
14.Z SCALE FACTOR -      -1.000
ENTER NUMBER OF PARAMETER TO CHANGE AND NEW VALUE (IF APPROPRIATE)
14,-5

   .OPTION = 1.
2.TITLE = FWING-RODY, 11 DEG, V=100 FPS (XWM4)    15:06:524-AUG-81
3.X AXIS ANGLE =       0.000
4.Y AXIS ANGLE =     120.000
5.Z AXIS ANGLE =      90.000
6.X MIN =      -2.000
7.X MAX =       8.000
8.Y MIN =      -2.000
9.Y MAX =       8.000
10. Z VARIABLE = 10. CTOTAL
14.Z SCALE FACTOR -      -5.000
ENTER NUMBER OF PARAMETER TO CHANGE AND NEW VALUE (IF APPROPRIATE)
4
DO YOU WANT A HARDCOPY PLOT? N
7 @$d @
        TTO  --  STOP
>PIP MENU.DAT/PU
>RUN [152,13]GRAFS
```

FWING-BODY, 11 DEG, V=100 FPS (XWM4)   15:06 24-AUG-81

CTOTAL

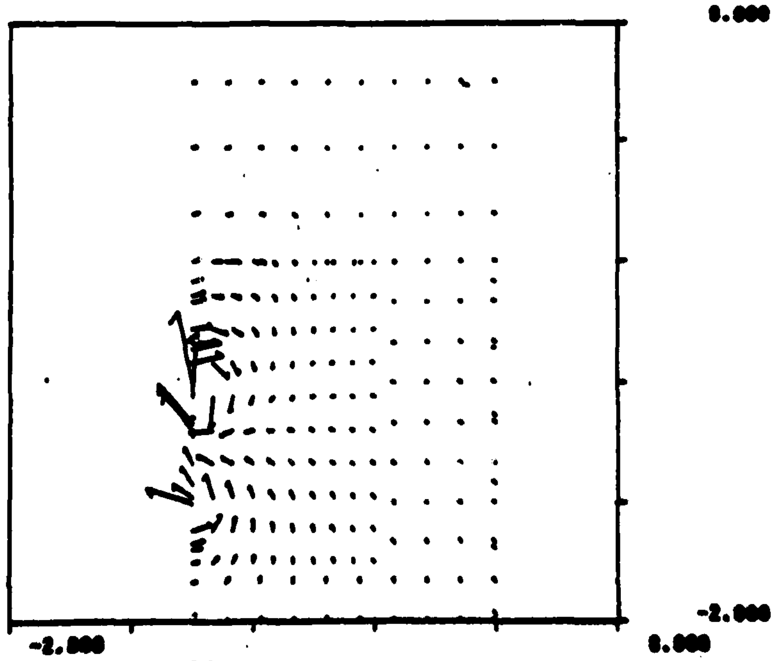


```
TTO   --  STOP
>* DO YOU WANT ANOTHER GRAPH? [Y/N]:Y


>* DO YOU WANT TO USE A DIFFERENT MENU FILE? [Y/N]: N
7RØN [152,13]MENU

   DO YOU WANT TO SEE THE DATA MINIMUM/MAXIMUM VALUES? [Y/N]
N



    .OPTION = 1.
2.TITLE = FWING-BODY, 11 DEG, V=100 FPS (XWM4)     15:06:524-AUG-81
3.X AXIS ANGLE =      80.000
4.Y AXIS ANGLE =      90.000
5.Z AXIS ANGLE =      90.000
6.X MIN =      -2.000
7.X MAX =       8.000
8.Y MIN =      -2.000
9.Y MAX =       8.000
10. Z VARIABLE = 10.
14.Z SCALE FACTOR -       -5.000
ENTER NUMBER OF PARAMETER TO CHANGE AND NEW VALUE (IF APPROPRIATE)
[illegible]
```

```
   .OPTION = 2.
2.TITLE = FWING-BODY, 11 DEG, V=100 FPS (XWM4)    15:06:524-AUG-81
3.X AXIS ANGLE =       0.000
4.Y AXIS ANGLE =      90.000
5.Z AXIS ANGLE =       0.000
6.X MIN =      -2.000
7.X MAX =       8.000
8.Y MIN =      -2.000
9.Y MAX =       8.000
10.VECTOR LENGTH SCALE FACTOR =       0.500
ENTER NUMBER OF PARAMETER TO CHANGE AND NEW VALUE (IF APPROPRIATE)
1001
```

```
   .OPTION = 2.
2.TITLE = FWING-BODY, 11 DEG, V=100 FPS (XWM4)    15:06:524-AUG-81
3.X AXIS ANGLE =       0.000
4.Y AXIS ANGLE =      90.000
5.Z AXIS ANGLE =       0.000
6.X MIN =      -2.000
7.X MAX =       8.000
8.Y MIN =      -2.000
9.Y MAX =       8.000
10.VECTOR LENGTH SCALE FACTOR =       1.000
ENTER NUMBER OF PARAMETER TO CHANGE AND NEW VALUE (IF APPROPRIATE)
[illegible]
DO YOU WANT A HARDCOPY PLOT? N
7 @@d @
       TTO  --  STOP
>PIP MENU.DAT/PU
>RUN [152,13]GRAFS
```

FWING-BODY, 11 DEG, V=100 FPS (XWM4)   15:06 24-AUG-81



```
TTO  --  STOP
>* DO YOU WANT ANOTHER GRAPH? [Y/N]: N
>* DO YOU WANT TO SAVE THE MENU DATA FILE? [Y/N]: N
>PIP MENU.DAT;*/DE/NM
>PIP DATAIN.DAT;*/DE/NM
>* DO YOU WANT TO PLOT ANOTHER FILE? [Y/N]: N
>@ <EOF>
>
```

APPENDIX C

DFAN Standard Data File Format

File structure: Formatted, sequential

Record 1: Format 5I5

5 Integer entries as follows:

1 - ND - Number of dimensions in a data point

2 - NP - Number of data points per line

3 - NL - Number of lines per grid

4 - NG - Grid number

5 - NLAT - Lattice number

Record 2: Format 20A1, 60A1

2 Logical*1 Arrays as follows:

1 - STITLE - 20 character short title

2 - LTITLE - 60 character long title

Characters 41-48 are Time-HH:mm:ss

Characters 49-60 are Date-DD-mmm-yy

Record 3: Format 10A1

ND Logical*1 Dimension Names/Labels

1 - ND - DLABEL - 10 Character Dimension Name/Label

Record 4: Format NP*NL+3: Format 10E12.5

ND Real numbers for a data point

1 - ND - DAT - Values for each dimension in a data point

This basic structure may be repeated as many times as necessary in a single file.

FILMED

8-8